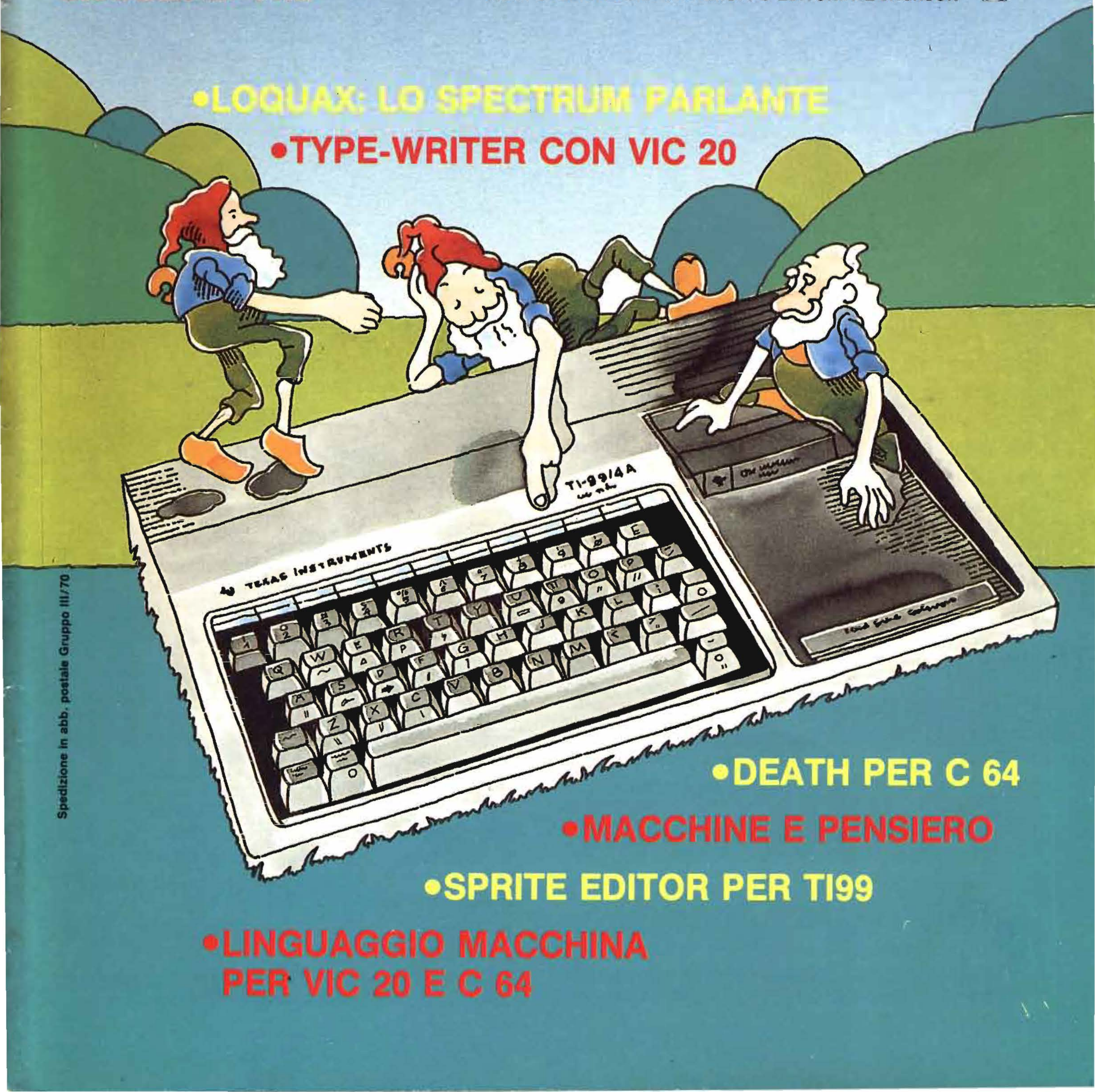
LA PRIMA RIVISTA EUROPEA DI SOFTWARE PER PERSONAL COMPUTER

ISSN 0392-8896

PERSONAL SOFTWARE

ANNO 3 N. 19
GIUGNO-LUGLIO 1984 - L. 4.000

UNA PUBBLICAZIONE DEL GRUPPO EDITORIALE JACKSON



- LOQUAX: LO SPECTRUM PARLANTE
- TYPE-WRITER CON VIC 20
- DEATH PER C 64
- MACCHINE E PENSIERO
- SPRITE EDITOR PER TI99
- LINGUAGGIO MACCHINA PER VIC 20 E C 64

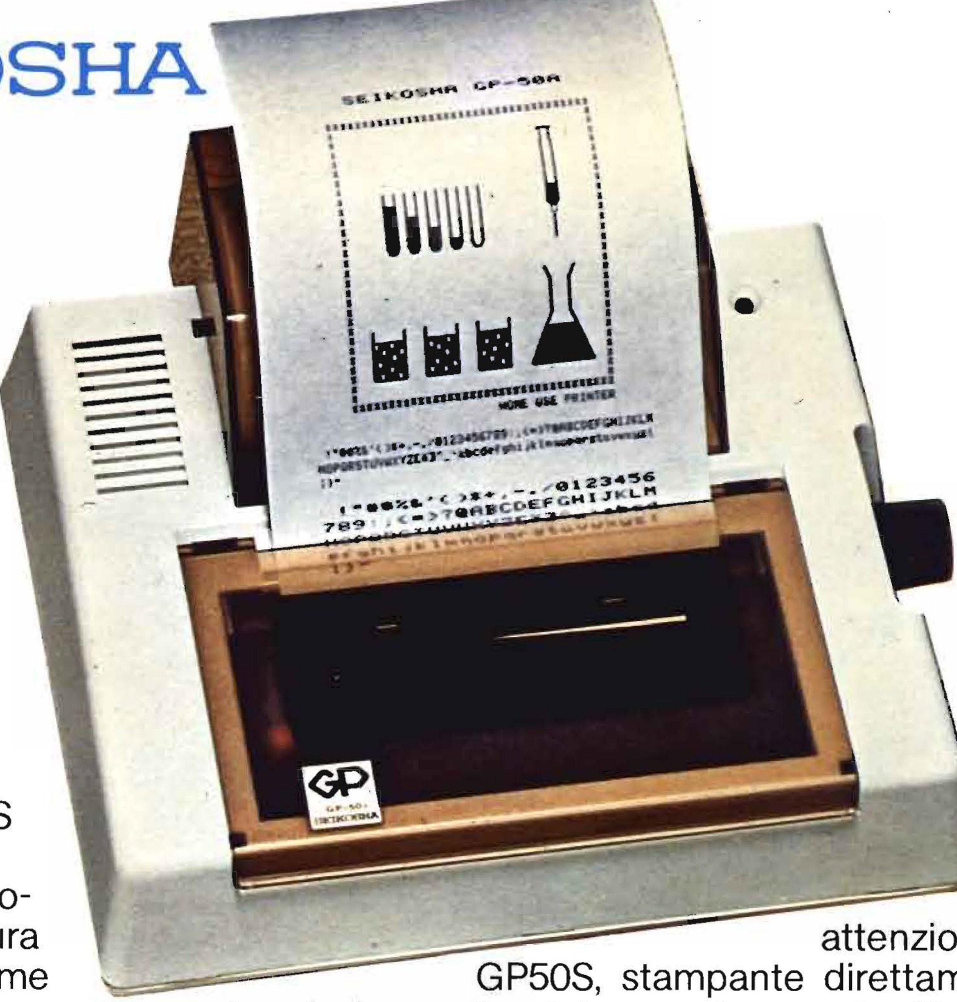
Spedizione in abb. postale Gruppo III/70

GP50A E GP50S

le piccole stampanti per tutti i computer

SEIKOSHA

REBIT COMPUTER
A DIVISION OF G.B.C



Piccole e compatte dalle prestazioni grandi e generose, le GP50A e GP50S sono realizzate con standard professionali a misura di Personal e Home computer e si impongono quale soluzione ottimale per gli usi hobbystici più disparati a costi incredibilmente sorprendenti. Particolare attenzione merita la GP50S, stampante direttamente interfacciata verso i computer Sinclair ZX81 e Spectrum.

Caratteristiche:

- Stampante ad impatto a matrice di punti da 46 colonne (32 colonne versione GP50S)
- Matrice di stampa 5x8 (7x7 versione GP50S)
- Percorso di stampa monodirezionale (da sinistra a destra)
- Capacità grafiche con indirizzamento del singolo dot
- Possibilità di ripetizione automatica di un carattere grafico
- Velocità 40 caratteri/secondo (35 caratteri/secondo versione GP50S)
- Caratterizzazione: 12 cpi e relativo espanso
- Interfacce: parallela centronics (interfaccia Sinclair versione GP50S)
- Alimentazione carta a frizione (largh. carta fino a 5")
- Stampa 1 originale e 1 copia
- Set di 96 caratteri ASCII
- Consumo 11W (standby) o 17W (stampa)
- Livello di rumore inferiore a 60 dB
- Durata di vita testa: 30 milioni di caratteri
- Peso 1,5 KG
- Dimensioni: 215 (prof.) x 250 (largh.) x 85 (alt.) mm.
- Nastro nero (standard); optionals: rosso, arancio, verde, blu, viola e marrone.



In copertina: questo mese tutti i segreti per animare gli "sprite" del TI99/4A.

N. 19
GIUGNO-LUGLIO 1984

PERSONAL SOFTWARE

ARTICOLI

GUIDA

RUBRICHE

CONVIENE ABBONARSI ....
.... ALLE RIVISTE JACKSON

Uno sconto sicuro per chi sottoscrive un abbonamento

Riviste	Uscite	Importo globale	Costo abbonamento	Costo abbonamento estero
Personal Software	11	~~L. 44.000~~	L. 34.000	L. 51.000
Bit	11	~~L. 55.000~~	L. 43.000	L. 64.500
Informatica Oggi	11	~~L. 38.500~~	L. 31.000	L. 46.500
Elektor	12	~~L. 36.000~~	L. 29.000	L. 43.500
Automazione Oggi	11	~~L. 38.500~~	L. 30.500	L. 45.250
Elettronica Oggi	11	~~L. 44.000~~	L. 36.000	L. 54.000
L'Elettronica	22	~~L. 55.000~~	L. 44.000	L. 66.000
Telecomunicazioni Oggi	10	~~L. 35.000~~	L. 28.000	L. 42.000
Video Giochi	11	~~L. 38.500~~	L. 30.000	L. 45.000
Strumenti Musicali	10	~~L. 30.000~~	L. 24.000	L. 36.000
Home Computer	11	~~L. 38.500~~	L. 31.000	L. 46.000

Un supersconto a chi si abbona a due o più riviste

Tutti coloro che sottoscrivono l'abbonamento a due o più riviste godono di un prezzo ulteriormente agevolato, come appare nella seguente tabellina.

Esempio: Bit + Informatica Oggi
(Italia) L. 43.000 + L. 31.000 = L. 74.000 **meno** L. 2.000 = L. 72.000!
(Estero) L. 64.500 + L. 46.500 = 111.000 **meno** L. 2.000 = L. 109.000

Abbonamento a 2 riviste:	L. 2.000 **in meno** sulla somma dei 2 prezzi d'abbonamento
Abbonamento a 3 riviste:	L. 4.000 **in meno** sulla somma dei 3 prezzi d'abbonamento
Abbonamento a 4 riviste:	L. 7.000 **in meno** sulla somma dei 4 prezzi d'abbonamento
Abbonamento a 5 riviste:	L. 10.000 **in meno** sulla somma dei 5 prezzi d'abbonamento
Abbonamento a 6 riviste:	L. 13.000 **in meno** sulla somma dei 6 prezzi d'abbonamento
Abbonamento a 7 riviste:	L. 16.000 **in meno** sulla somma dei 7 prezzi d'abbonamento
Abbonamento a 8 riviste:	L. 20.000 **in meno** sulla somma degli 8 prezzi d'abbonamento
Abbonamento a 9 riviste:	L. 25.000 **in meno** sulla somma dei 9 prezzi d'abbonamento
Abbonamento a 10 riviste:	L. 30.000 **in meno** sulla somma dei 10 prezzi d'abbonamento
Abbonamento a 11 riviste:	L. 35.000 **in meno** sulla somma degli 11 prezzi d'abbonamento

Per abbonarsi

Per sottoscrivere abbonamenti utilizzate il modulo di c.c.p. n. 11666203 intestato a Gruppo Editoriale Jackson - Via Rosellini, 12 - Milano, oppure inviate un assegno o un vaglia postale al nostro ufficio abbonamenti.

EDITORIALE

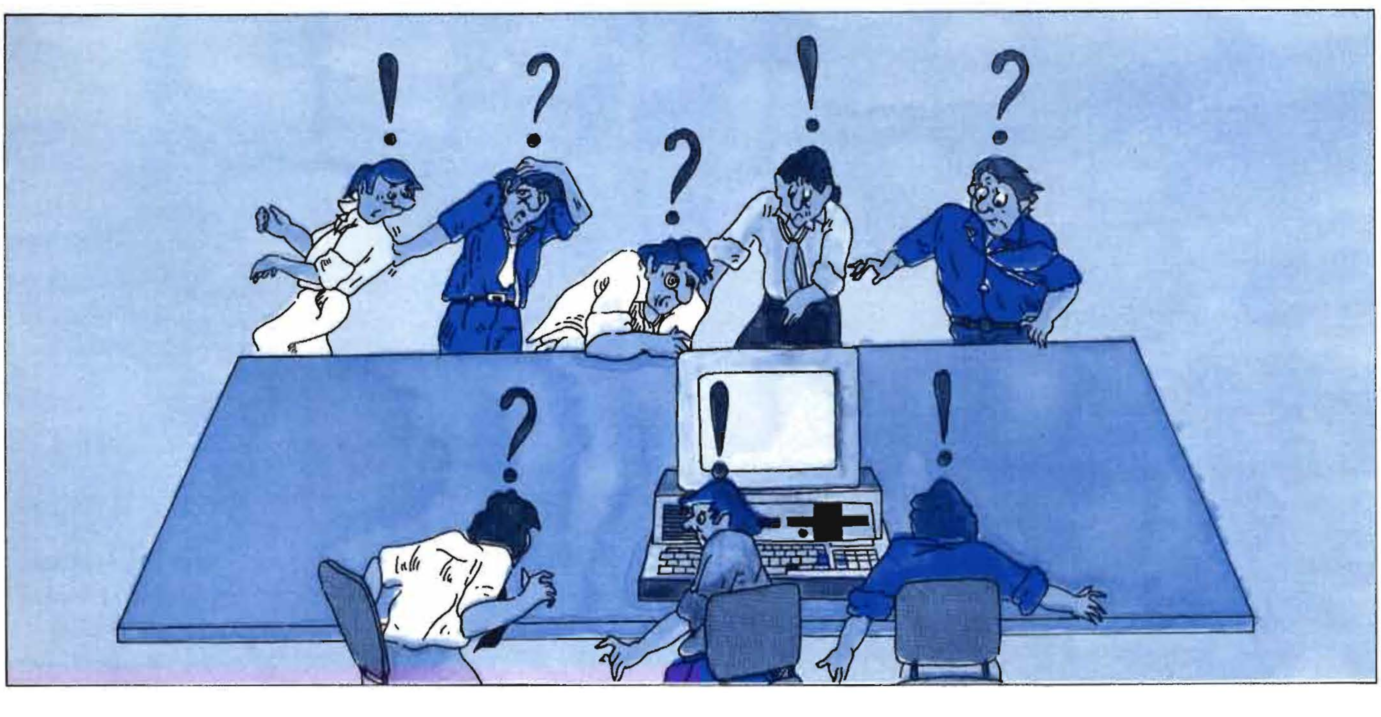
Le domande imbarazzanti

di Riccardo Paolillo

Ogni volta che mi capita di parlare di personal computer con conoscenti alle prime armi o completamente al di fuori dal mondo della microinformatica, so già che prima o poi mi arriverà, inesorabile, una domanda diretta e precisa che, lo confesso, mi mette regolarmente in crisi: "Ma, in definitiva, cosa si può fare con un personal computer?".
In questi casi balbetto qualche risposta, parlando di semplici e spesso improbabili mini-applicazioni, inventate sul momento, cercando di utilizzare un linguaggio il più piano possibile.
Il risultato inevitabile che ottengo e che leggo negli occhi dei miei interlocutori è sempre sconfortante: i dubbi sono rimasti, quelle che a me sembrano problematiche molto importanti per altri sono del tutto irrilevanti.

Morale: non si può pretendere di spiegare, in pochi minuti e solo a parole, l'uso e le possibilità di impiego di uno strumento potente e per molti ancora sconosciuto, quale è il personal computer. Per conoscere ed apprezzare il calcolatore occorre toccarlo con mano, usarlo, anche se inizialmente questo può costare qualche notte insonne e parecchie maledizioni lanciate alla macchina.
Secondo me il difetto di molti di noi computermaniaci, appassionati di personal e di programmazione, è quello di pensare di poter trasmettere in modo automatico e veloce il nostro entusiasmo a chiunque ci faccia qualche innocente domanda sulla microinformatica. In realtà, in questo campo, il contagio può essere aiutato, ma alla base deve esistere un reale interesse da parte di chi si vuole introdurre al mondo dei personal.
Infatti non tutti sono ugualmente sensibili davanti a quelle che a noi sembrano delle prestazioni eccezionali dei nostri calcolatori personali: a chi, mostrando con orgoglio il proprio sofisticato programma grafico, non è capitato di sentirsi dire che sì, è bello, ma le rette sono in realtà delle spezzate e i cerchi non sono poi così rotondi?
Oppure, dopo aver fatto ascoltare la routine sonora costata tanta fatica, suscitare commenti del tipo: "Il suono è troppo gracchiante, qualunque organino a buon mercato può fare di meglio".
D'altra parte il mondo è bello nella sua varietà: non si può certo pretendere che tutti apprezzino e capiscano i personal computer.

Ma per tutti gli altri, per quelli che invece si interessano a questo nuovo fenomeno? Mi rendo conto che ancora una volta sto cercando di schivare la domanda iniziale sulle possibilità di utilizzo. Potrei rispondere, restando in questo tipo di logica necessariamente utilitaristica, elencando i più comuni campi di impiego: musicale, grafico, memorizzazione dati personali, scrittura di testi, videogiochi, ecc..
Sono tutte possibilità serie e molto interessanti, eppure preferisco dire: provatelo, magari senza decidere a priori cosa farne. Scopritelo poco alla volta e vedrete che entro pochissimo tempo troverete da soli tantissime possibilità di impiego utili o anche, perché no, solamente divertenti. ■



POSTA

Varie Spectrum

Sono uno studente possessore di uno Spectrum 48 Kbyte.
Mi sono abbonato a **Personal Software** per l'alta qualità dei suoi articoli, riguardanti i personal in generale, ma soprattutto, naturalmente, per quelli targati Spectrum.
Mi rivolgo a voi per alcune domande, probabilmente banali, ma sono un dilettante.
1) Le periferiche da collegare allo Spectrum, vanno poste sotto tensione dopo l'accensione dello Spectrum.
E il televisore? (ho messo un interruttore che li accende contemporaneamente. Ho fatto male?).
2) Digitando PRINT PEEK 23732 + 256 ★ PEEK 23733 ottengo 65535. Mi hanno detto che dovrei ottenere 65536. Come mai? Aggiungo che il mio Spectrum, anche se acquistato il 23/12/83, è della produzione 1982.
3) Quando carico un programma i beep sono molto alti come volume. Ho notato che altri Spectrum sono più silenziosi. D'altra parte, se abbasso il volume del tape lo Spectrum mi dà errore. È normale?
4) Potreste pubblicare per intero il mio indirizzo perché in questa zona non riesco a trovare altri Spectromani.

Alessandro Paroli
Via 2 Giugno, 2
57020 La California (LI)

È assolutamente irrilevante l'ordine con cui vengono messi sotto tensione televisore, Spectrum e registratore a cassetta. La sua soluzione va senz'altro bene, se ha avuto l'accortezza di sistemare l'interruttore in modo da escludere, in posizione off, l'alimentatore dello Spectrum, per non surriscaldarlo inutilmente.
Le locazioni 23732 e 23733 contengono l'indirizzo dell'ultimo byte di RAM fisica. I byte indirizzabili dal microprocessore Z80 sono 2 ↑ 16 = 65536, numerati da 0 (anziché 1 come ci verrebbe spontaneo pensare) fino a 65535. Ecco quindi spiegato il piccolo mistero del byte mancante.
La terza domanda ci sembra un po' strana: probabilmente il registratore che usa è differente da quello adoperato con altri Spectrum.
Infine, siamo lieti di pubblicare il suo indirizzo completo anche se, ci scusi la facile ironia, stupisce una simile richiesta proprio dalla California ...

■ ■ ■

Abukir per C 64

In riferimento al listato del programma "Abukir 1798", pubblicato sul n. 16 della vostra rivista, vi facciamo notare che il suddetto listato digitato sul Commodore 64 fino alla linea 2043 provoca degli errori (?SYNTAX ERROR) nelle linee da 85 a 185 corrispondenti ai "DATA".
Desidereremmo quindi avere, se possibile, alcune spiegazioni in merito affinché il programma possa girare senza errori.

Gianni Lucatorto
Marco De Benedetto
Bari

Il fatto che il programma Abukir 1798 non funzioni su C 64 è perfettamente comprensibile, dato che è stato scritto per i PET/CBM vecchie serie. Visto il notevolissimo interesse mostrato da parecchi utenti di C 64, abbiamo chiesto all'autore del gioco, Umberto Barzaghi, di preparare una versione anche per questa macchina. Umberto, nonostante i numerosi impegni, si è messo all'opera e quindi prevediamo di pubblicare il listato in tempi brevissimi.
Lo stesso procedimento, cioè la traduzione per C 64, verrà effettuato quanto prima anche per il successivo gioco dello stesso autore, "La battaglia del lago ghiacciato", pubblicato nei numeri 17 e 18.
A tutti i sessantaquattristi impazienti, che presto saranno soddisfatti, ricordiamo che ci sono anche i possessori dei vecchi PET/CBM da accontentare.

■ ■ ■

Qualche critica (e richieste)

Sono un lettore della vostra rivista e vorrei esporvi alcune lamentele.
Vi anticipo che lo scopo di questa mia lettera ha esclusivamente fini costruttivi e vi prego di non voler intendere alcuni passaggi come esageratamente polemici.
Fin dal primo numero ho trovato difficoltà a reperire i numeri della vostra rivista e, nonostante chieda in continuazione a vari giornalai notizie sull'eventuale uscita, ho già perso due numeri della vostra rivista (numeri 1 e 15).
Il numero di riviste che voi distribuite è talmente irrisorio che sono molti coloro che restano a bocca asciutta.
A questo punto potreste rispondermi che l'unico modo di ovviare a questo problema è di abbonarmi alla vostra rivista e di richiedere i numeri arretrati pagando il doppio del prezzo di copertina.
Ma io ritengo che questa vostra ipotetica risposta sia un'imposizione nei riguardi degli acquirenti di riviste di informatica.
Con l'abbonamento si viene a perdere la possibilità di poter scegliere quale rivista più interessa, costringendo noi lettori a comprare a scatola chiusa.
D'altro canto non sono disposto a pagare a mie spese quella che io ritengo una vostra manchevolezza; mi sembra quindi ingiusto chiedere di

POSTA

pagarvi 7.000 lire per un arretrato perso per colpa vostra.
Passo ora alla seconda parte delle mie critiche.
Inizialmente la vostra rivista prometteva molto, illustrando metodologie di programmazione e programmi illustrati, senza far riferimento a specifici calcolatori.
Negli ultimi numeri va sempre più affermandosi, purtroppo, la tendenza a riportare programmi già sviluppati per i computer più diffusi.
Io posseggo un home computer (Newbrain A) poco diffuso sul mercato e non posso usufruire dei programmi da voi pubblicati.
Io penso che i veri appassionati dell'arte di programmare, non i fanatici dei video games e programmi pronti per l'uso, gradiscano aggiornarsi sulle metodologie di programmazione, che sono molto più interessanti (ve lo dico per esperienza essendo iscritto al V anno di ing. elettronica) di un semplice programmino pronto per l'uso. Spero quindi in un vostro ritorno alle origini.

(Lettera firmata)

L'autore di questa lettera ci ha chiesto, chissà perché, una risposta privata.
Noi ne rispettiamo l'anonimato, ma preferiamo rispondere sulla rivista perché riteniamo questi discorsi di interesse generale.
Il problema della distribuzione capillare di una rivista come **Personal Software** *su tutto il territorio nazionale non è ovviamente semplicissimo.*
Nei mesi scorsi il distributore nazionale, che non è il Gruppo Jackson, ha incontrato parecchi problemi, in quanto la richiesta della rivista è andata aumentando ogni numero. Noi abbiamo costantemente aumentato la tiratura e crediamo che ora il problema sia in gran parte risolto.
Le assicuriamo che non abbiamo mai pensato di creare ad hoc queste situazioni per incrementare gli abbonamenti o vendere gli arretrati. Infatti il problema è solo tecnico: la mancanza di copie della rivista in determinate zone può essere rilevata dal distributore solo il mese successivo.
Per quanto riguarda la seconda parte della lettera, riprendiamo un discorso già fatto più volte, ma che trattiamo sempre volentieri, in quanto ci permette di conoscere sempre di più le esigenze e le preferenze dei lettori.
Dato per scontato che il partito degli appassionati a programmi di "consumo" (videogame e affini) è sempre forte e compatto, gradiremmo conoscere anche le opinioni dei lettori che preferiscono articoli di altro tipo.
Ci piacerebbe ricevere proposte concrete e sulla base delle richieste pervenute cercheremo senz'altro di accontentarvi.
Comunque, anche negli ultimi numeri ci sembra di aver affrontato problematiche che vanno al di la del semplice gioco preconfezionato.

PERSONAL NEWS

Commodore 64: che musica, ragazzi

I primi timidi "bip" emessi faticosamente dai personal di qualche anno fa, la tecnologia dei generatori di suono per i microcomputer ha fatto passi da gigante, fino ad arrivare a permettere l'inserimento di un vero e proprio sintetizzatore a tre oscillatori in uno degli apparecchi più popolari e diffusi, il Commodore 64, appunto. L'architettura di questo sintetizzatore è molto simile a quella di uno strumento "storico" per la musica pop e rock, il Minimoog. Il



nostro generatore di suoni è però completamente programmabile ed è contenuto in un chip non più grande di un francobollo, il SID (Sound Interface Device) 6581. Al posto dei tasti e degli interruttori di un tradizionale sintetizzatore, il SID contiene dei registri di controllo accessibili da BASIC tramite l'istruzione POKE che consente di comandare tutti gli effetti sonori possibili. Dopo aver acquistato un po' di confidenza con il BASIC e con le convenzioni relative al significato dei bit di controllo dei registri del SID, si può cominciare a lavorare con questi "interruttori virtuali" come se fossero i tasti e il pannello di un comune sintetizzatore. È possibile, ad esempio, formare accordi di tre note, modulare scale cromatiche più o meno temperate, esibirsi in valzer lenti o inventare i ritmi più strampalati, a patto di aver preso dimestichezza con lo "strumento". Esistono in commercio, per i pigri..., programmi che consentono di utilizzare la tastiera del C 64 come se fosse quella di un pianoforte (non cominciate a dipingere di nero i tasti!) o di vedere sul pentagramma che compare sullo schermo la melodia che si sta eseguendo. Alcuni lo hanno già utilizzato per creare opere di videoart sonorizzata, a Reggio Emilia stanno componendo un pezzo per banda e computer, qualcun'altro progetta di collegarlo ad altri strumenti elettronici. Non c'è male, soprattutto se si considera che il Commodore 64 costa meno di un sintetizzatore!

Commodore S.r.l.
Via Fratelli Gracchi, 48
Cinisello Balsamo (MI)
Tel. 02-6125651

Computer Club a Grugliasco

Un effervescente Computer Club ci prega di dare spazio alle sue iniziative. Lo facciamo volentieri, ricordando a tutti i club che siamo disposti a pubblicare comunicati di interesse generale. A Grugliasco il locale Computer Club organizza corsi introduttivi e di BASIC; grazie al successo anche esterno dell'iniziativa (arrivata a proporre dei corsi per i CRAL di Torino con pratica su macchine Apple) per il prossimo autunno sono in cantiere corsi di Pascal, COBOL e Assembly: vale la pena di prenotarsi. L'associazione al Club (costo: 5000 lire) dà diritto ad un elenco di programmi disponibili per l'acquisto a prezzi contenuti, con particolare ricchezza di software per Spectrum, Apple e VIC. Per informazioni telefoniche, farsi vivi il lunedì in orario d'ufficio allo 011-7803674.

Computer Club Il Laboratorio
Via Cotta, 20
10095 Grugliasco (TO)

Registrateli su cassetta

Arriva da Taiwan l'annuncio di un convertitore che permette di memorizzare su un'ordinaria C-60 fino a 8 giochi Atari. Questo caricatore, detto Wizard Charger MG-400, viene inserito nel Video Computer System mentre dall'altro lato riceve la video cassetta. Agendo opportunamente sui tasti dell'Atari è possibile trasfe-



rire il video gioco sul registratore a cassette inserito nel Wizard Charger. È naturalmente possibile anche leggere il video gioco direttamente dalle audio cassette preparate con questo sistema. Questo prodotto (che non crediamo faccia fare salti di

gioia ai produttori e ai rivenditori ufficiali di materiale originale) consentirà di registrare e scambiare video giochi con la stessa facilità con cui si riproducono e ci si scambiano le cassette di musica pop e rock.

MITAC Inc.
Nanking E. Road
Taipei, Taiwan, R.O.C.

Dragon 32 raddoppia e arriva il 64

La Dragon, distribuita in Italia dalla ECO di Verona, ha messo a punto una versione da 64 Kbyte di memoria del suo home computer, dotandolo di software applicativo e di unità a dischetti che ne rendono possibile l'utilizzo anche per attività professionali. La nuova macchina, detta Dragon 64, presenta tre "modi" principali di operazione: è completamente compatibile con il precedente 32 nella modalità 32 Kbyte. Per l'uso dell'interprete BASIC Microsoft è disponibile la modalità 48 Kbyte, che vede impegnati 16 Kbyte di RAM per l'interprete e ne lascia 48 per i programmi utente. Esiste anche un sistema operativo, l'OS-9, capace di utilizzare completamente i 64 Kbyte di memoria a disposizione nell'ultima modalità consentita. A differenza di molti altri elaboratori "a la page", il Dragon 64 non utilizza un processor a 16 bit, ma monta, come il 32, un 6809E a 8 bit. La tastiera è professionale, QWERTY a 53 tasti, le porte previste sono una RS 232 seriale, una parallela Centronics e le uscite per 2 joystick, per il video e il registratore a cassette. Il disk drive (a "mezza altezza") tratta



floppy a singola faccia e doppia densità, per un totale di 180 Kbyte visibili all'utente. Il sistema operativo per l'uso dei dischi è residente nella ROM presente sulla scheda di controllo del drive. Sono previsti, oltre al citato Microsoft BASIC, il Pascal, l'Assembly e il linguaggio C. I prezzi? Il Dragon 64 costa 690.000, il controller per le unità a disco 140.000, il cabinet con un drive 610.000 mentre la versione con due drive ha un prezzo di 990.000 lire. Il software a disposizione costa dalle 17.800 lire per i giochi alle 160.000 del Super Writer II.

ECO S.r.l.
Via Prato Santo, 18
37126 Verona
Tel. 045-913297

S.I.EL.

Abbiamo deciso di dare carattere sistematico e continuativo ad un canale d'informazione diretto ai nostri vecchi e molti amici: i musicisti. Ed abbiamo anche deciso di attivare un altro canale diretto d'informazione, anch'esso permanente, rivolto ai nuovi amici che andiamo cercando tra gli appassionati di home computer e tra tutti coloro che, per hobby o per professione, s'interessano di informatica. Molti di voi sono sicuramente anche appassionati di musica e noi, che siamo esperti nel settore dello strumento musicale e siamo entrati a vele spiegate nel settore dell'informatica applicata alla musica, pensiamo di avere per voi proposte interessanti ed informazioni sempre fresche per mantenervi aggiornati.
Per realizzare la nostra iniziativa abbiamo scelto quattro testate tra le più qualificate nell'area della stampa specializzata: *Strumenti Musicali, Bit,* **Personal Software** e *HC.*
Tutte e quattro, sono testate del Gruppo Editoriale Jackson....
E non è un caso... loro specializzati, noi anche: cercarsi, trovarsi e collaborare è consequenziale.
Lasciateci ora un po' di spazio per dirvi chi siamo e, prima ancora, cosa abbiamo fatto per arrivare ad essere quel che siamo.
I nostri vecchi amici - i musicisti - già sanno chi siamo e come ci muoviamo nel settore: molti di loro hanno certamente utilizzato le nostre tastiere, i nostri piani elettronici e parecchi sono già ora alle prese con il sintetizzatore polifonico Opera 6, con il suo Expander e la relativa interfaccia MIDI.
E già siamo nell'informatica musicale: già strumenti musicali e personal computer si sposano promettendo miracoli!
E già stiamo parlando ai nostri nuovi amici di come essi potranno - finalmente - far musica sul serio servendosi dei loro personal computer ma... servendosi di strumenti musicali di qualità elevata dalle timbriche raffinate ben al di sopra delle prestazioni musicali, sempre accessorie, degli home computer.
La nostra ambizione è quella di fare degli informatici dei patiti della musica e viceversa! E siamo convinti di farcela.

PERSONAL NEWS

Ma torniamo a noi ed alla nostra breve storia.
La S.I.EL. in versione 1984 dispone di uno stabilimento di circa 8.000 mq., occupa 130 persone di cui oltre il 15% (20 tecnici) sono addetti al reparto ricerca e sviluppo.
Già l'organico del reparto ricerche, se rapportato all'organico complessivo dell'unità produttiva, è un chiaro indice delle metodologie operative della nostra azienda. Ciascun nuovo progetto, scaturito da una accurata ricerca di marketing, coinvolge, globalmente, tutte le componenti del nuovo prodotto: parti elettroniche, elettromeccaniche e meccaniche; senza dimenticare funzionalità, praticità d'impiego ed estetica.
Anche l'ingegnerizzazione di ogni nuovo prodotto occupa un posto importante nelle procedure aziendali ed è seguita, nella fase produttiva, da un rigido controllo qualità espletato internamente per i materiali in ingresso e per i semilavorati eventualmente prodotti all'esterno. Così otteniamo l'elevato standard qualitativo dei nostri prodotti.
Il livello dei nostri tecnici è ottimo ed è parte della politica aziendale un loro costante aggiornamento. Tra di essi sono presenti anche tecnici/musicisti, capaci quindi di interpretare le esigenze dell'utilizzatore finale e questo ci garantisce la capacità di adeguarci con esattezza e sufficiente elasticità ai mutevoli indirizzi del mercato.
La nostra produzione è indirizzata per l'80% all'esportazione. Vendiamo in tutti i Paesi dell'Europa occidentale, nei Paesi del Medio Oriente e negli Stati Uniti. Ovunque disponiamo di una rete distributiva capillare, in alcune aree controllata direttamente dalla S.I.EL. stessa.
L'organizzazione interna dell'azienda è interamente controllata da un IBM 38: programmazione delle commesse, degli approvvigionamenti, della produzione, del magazzino in tutti i suoi settori, amministrazione e contabilità generale, tutto fa capo al centro meccanografico naturalmente accessibile anche ai progettisti.
La storia di un'azienda, specie se letta dall'utilizzatore, si identifica nei suoi prodotti: ricordiamone alcuni tra i più recenti e significativi, indicandone gli anni di presentazione.
Cruise 5 (1980) Orchestra 1. (1982).
Il primo è una tastiera dotata di effetti quali archi, brass e strumenti percussivi, completato da un sintetizzatore monofonico.
Il secondo è un'altra tastiera, fornita di timbri simili al precedente ma senza sintetizzatore che è la versione aggiornata di un modello similare



presentato circa due anni prima.
Cruise ed Orchestra sono strumenti dedicati al "tastierista" che deve amalgamare gli strumenti (solitamente pochi) del suo complesso con sonorità orchestrali. L'obiettivo è stato pienamente centrato e possiamo dire di aver coperto con questi due strumenti (tre considerando la prima versione dell'Orchestra) un'area di mercato trascurata da molti.
E la ragione del successo sta in quel "punto di equilibrio" di cui abbiamo detto.
Negli anni 1982/83 abbiamo dedicato la nostra attenzione anche agli amatori, ai principianti che cercano una via facile per avvicinarsi alla musica. È nato così il Flying in due versioni contraddistinte dai numeri 49 e 61 indicanti il numero dei tasti.
Il Flying è dotato di una batteria elettronica e di tutta una sezione di accompagnamenti automatici per ottenere gli accordi corretti nel tempo corretto: proprio ciò che occorre ad un principiante.
Nella progettazione di questo strumento è stata dedicata molta cura anche alla linea, visto che si trattava di un italiano, carrozzato in Italia, che doveva combattere contro l'agguerrito invasore giallo che ha profuso in questo settore investimenti ed energie a non finire. E dal confronto il Flying non è affatto uscito con i tasti pesti!
Poi sono arrivati i piani elettronici Piano 4 e PX con dinamica di tastiera, realizzati con la particolare tecnica della modulazione di ampiezza (1983). Molto potremmo dirvi di questi, ma il foglio bianco si accorcia inesorabilmente... passiamo subito agli ultimi arrivati.

Opera 6. Un sintetizzatore polifonico, con timbri programmabili, il nuovo strumento per i tastieristi che tiene validamente testa agli "invaders" d'ogni colore. È provvisto di MIDI, è cioè predisposto per dialogare con altri strumenti dotati di standard MIDI o per essere interfacciato con un personal computer! E per ora vi lasciamo immaginare tutto quello che si può fare... ve lo diremo la prossima volta!
Expander. E4 in pratica un Opera 6 senza tastiera che espande appunto le prestazioni del primo utilizzando quel protocollo MIDI al quale abbiamo accennato.
Opera 6 ed Expander uguale due strumenti (ma potrebbero anche essere una mezza dozzina) suonati da un solo esecutore.
Poi abbiamo preparato un accessorio: il MIDI Interface. È ciò che occorre per collegare l'Opera 6 ed il suo Expander ad un personal computer e ricavarne un potente sequencer, una macchina per comporre musica, un compagno per suonarci.
Non meravigliatevi troppo se parliamo così bene dei nostri strumenti....

se non ne fossimo convinti non li avremmo costruiti, non vi pare? Poi noi ci fidiamo del responso del mercato che è sempre il giudice d'appello: non ci ha ancora né contraddetti né delusi.
Quindi noi continuiamo per la nostra strada che mira decisamente ad una intensa utilizzazione dell'informatica in musica, a favorire l'abbinamento strumento musicale/personal computer.
E proseguendo per questa strada stiamo preparando grosse sorprese... che non mancheremo di segnalarvi nei prossimi mesi.
Aspettatevi comunque qualche nuova sigla - per esempio MK - e vedremo insieme cosa nasconde!

S.I.EL. S.p.A.
Zona Industriale
63030 Acquaviva Picena (AP)
Tel. 0735-60744

FORTH Group

Si è costituito il FORTH Interest Group Italia, una associazione culturale senza fini di lucro legata al FORTH Interest Group statunitense, avente come fine lo scambio di informazioni ed esperienze tra utilizzatori del linguaggio FORTH.
I soci ricevono un notiziario bimestrale di oltre 40 pagine, contenente articoli, molti esempi di applicazioni, note sulle ultime novità relative al mondo FORTH ed il calendario di incontro per i soci.
L'associazione mette a disposizione dei soci vario materiale ed i listati sorgente del FORTH per quasi tutte le CPU usate a prezzi di costo.
L'iscrizione al gruppo dura un anno a partire da Giugno e per l'anno 1983/1984 la quota associativa è stata fissata in L. 40.000.
Per informazioni ed iscrizioni scrivete a:

FORTH Interest Group Italia
Via G. Forni, 48
20161 Milano

PERSONAL SOFTWARE

Controllo del video dal linguaggio macchina

Come sfruttare le routine della ROM per stampare sullo schermo dello Spectrum

di *Walter Sini*

Certo sono già molti i possessori di uno Spectrum che cominciano a divertirsi con piccole routine in linguaggio macchina. Pasticciando con PEEK e POKE si fa presto ad accorgersi che scrivendo "qualcosa" dall'indirizzo 16384 (4000H) all'indirizzo 22527 (57FFH), quel "qualcosa" appare sul video. Bene, abbiamo scoperto il Display File; di qui a voler visualizzare una lettera il passo è breve. Accendiamo il nostro beniamino e digitiamo POKE 16384, 255; ora cerchiamo il byte immediatamente sottostante: non è il 16385. È invece 16384 + 256; e questo per altre sei volte per poter completare la casella posta alla linea 0, colonna 0.

Il Display File è difatti organizzato in modo apparentemente strano, e occorre tener conto di questo per creare una routine di PRINT. Ma perché sprecare fatica?

Sicuramente da qualche parte nella ROM esiste già una routine di PRINT, basta trovarla. Ed allora via a spulciare riviste, cercando l'indirizzo di chiamata. Chi è fortunato scopre questa "formula magica":

ld a, N dove N è il codice del carattere da stampare;

rst 16 - 16 decimale o 10 esadecimale

ret.

Per caricarla usiamo il programmino del listato 1.

Dopo il RUN comparirà una "W" (CHR$ 87) nelle due righe riservate al computer, infatti, in mancanza di ulteriori istruzioni, è stata stampata nella stessa posizione dell'ultima stampa sul video, quella della parola RUN nella parte bassa del video.

Per stampare la nostra "W" nella parte superiore dello schermo è sufficiente aggiungere la linea "45 PRINT" al programma. Infatti il computer conosce almeno tre canali di output: la parte inferiore dello schermo, quella superiore e la stampante. Per indirizzarlo opportunamente occorre aprire il canale di comunicazione desiderato prima del comando di stampa.

I numeri di canale sono: 1 per la parte inferiore dello schermo, 2 per la parte superiore, 3 per la stampante.

In Assembly i canali si aprono così:

ld a, 2 - 2 per la parte superiore dello schermo

call 5633 - 1601h, è l'indirizzo di una routine che apre i canali

ld a, N - stampa

rst 16

ret.

Il nostro scopo è di usare esclusivamente l'Assembly per gestire il video, quindi cancelliamo la linea 45 appena aggiunta e sostituiamo il "3" alla linea 20 con "8" ed aggiungiamo la linea 55 DATA 62, 2, 205, 1, 22. Ora diamo il RUN. La "W" comparirà in alto a sinistra.

Il passo successivo è di stampare in una posizione a scelta. In Assembly si ottiene con:

ld a, 2 - parte superiore del video

call 5633 - apre il canale

ld b, L - L è il numero di linea

ld c, C - C è il numero di colonna

call 3545 - 0DD9h, è una routine che aggiorna le variabili di posizione

ld a, N - codice carattere

rst 16 - stampa

ret - ritorna al BASIC.

Per veder girare felicemente il tutto sostituiamo l'"8" alla linea 20 con "15" ed aggiungiamo la linea "57 DATA 6, <u>15</u>, 14, <u>15</u>, 205, 217, 13". Quelli sottolineati sono L e C che possono essere modificati a piacere, ma *attenzione*! Le linee sono numerate dal basso verso l'alto, dal nume-

```
10 CLEAR 59999: LET mem=60000
20 FOR a=0 TO 3
30 READ n: POKE mem+a,n
40 NEXT a
50 RANDOMIZE USR 60000
51 PAUSE 0
60 DATA 62,87,215,201
```

Listato 1. *Programma che effettua la chiamata alla routine di stampa dal linguaggio macchina. Per 16 Kbyte modificare i "60.000" in "30.000".*

```
55 DATA 62,2,205,1,22
57 DATA 6,24,14,33,205,217,13
58 DATA 33,143,92,54,130
60 DATA 62,87,215,201
```

Listato 2. *Insieme delle linee data con tutte le aggiunte.*

Controllo del video dal linguaggio macchina

ro 3 al numero 24 inclusi, le colonne da sinistra verso destra dal numero 33 al numero 3 inclusi. Per esempio, la casella in alto a sinistra ha le coordinate L24, C33.
Ma ... il colore? Per il colore c'è solo l'imbarazzo della scelta. È sufficiente aggiungere, *prima* della stampa, queste due istruzioni:
ld hl, 23695 - (questa è ATTR T, una variabile di sistema: vedi p. 176 del manuale)
ld (hl), Q.
In altre parole (o numeri) si aggiunge la riga "58 DATA 33, 143, 92, 54, Q" e si sostituisce "15" con "20". Vedi listato 2.
Ed ora parliamo di Q. Q è un numero dato dalla somma dei parametri che potete trovare nella tabella 1.
Esempio: per avere una lettera rossa su sfondo nero lampeggiante:
ink red = 2
paper black = 0
flash si = 128
Quindi: Q = 2 + 0 + 128 = 130.
Provare per credere. ■

COLORE	INK	PAPER
white	7	56
yellow	6	48
cyan	5	40
green	4	32
magenta	3	24
red	2	16
blu	1	8
black	0	0

ATTRIBUTO	SI	NO
flash	128	0
bright	64	0

Tabella 1. *Valori per selezionare gli attributi. È necessario sommare il valore scelto per ogni attributo.*

Lista delle variabili per Spectrum

Una routine da tenere sempre in memoria durante la messa a punto dei programmi

di *Raffaele Gallieni*

Ecco un'utilissima routine talvolta compresa nei "toolkit" (cioè i programmi che aggiungono nuovi comandi al computer con lo scopo di facilitare il lavoro del programmatore) dei personal più costosi.

Quando ci si trova in fase di messa a punto (debug) dei programmi è spesso utile potere prendere rapidamente visione del valore delle variabili contenute in memoria senza dover eseguire una noiosa serie di PRINT. La routine pubblicata svolge questa funzione: è sufficiente caricarla in qualunque momento con MERGE (non con LOAD, altrimenti si cancella il programma!) ed eseguire GOTO 9800 (non RUN 9800, altrimenti si cancellano le variabili!) ogni volta che si vogliono visualizzare i valori delle variabili.

Commenti al listato

Per non "sporcare" le variabili del programma principale ho usato poche variabili ed ho fatto grande uso di PEEK, ma questo ha compromesso la leggibilità.

Il ciclo principale (linee 9800 - 9834) per prima cosa individua le locazioni di inizio e di fine dell'area delle variabili di sistema, puntate rispettivamente da VARS (23627/8) e da E LINE (23641); il ciclo termina se la variabile di lavoro "v" supera E LINE o se viene trovato un CHR$ 128 (il che è la stessa cosa).

Quindi (linee 9812 - 9822) viene stabilito il tipo di variabile, esaminando i bit 7, 6 e 5 del primo byte della variabile, secondo il seguente schema:

bit7	bit6	bit5	descrizione
0	1	1	var. numerica, identificatore a 1 lettera
1	0	1	var. numerica, identificare a più caratteri
1	1	1	var. di controllo di un ciclo FOR NEXT
0	1	0	stringa di caratteri
1	0	0	matrice numerica
1	1	0	matrice di caratteri

Al termine del ciclo viene presentata l'opzione di stampa (linea 9830) che, in caso di risposta affermativa apre il flusso 2 (la stampante) e ripete l'esecuzione.

Nelle linee 9836 - 9922 si analizzano le variabili, stampando prima gli identificatori, poi il contenuto e incrementando opportunamente la variabile di lavoro "v" per l'iterazione seguente.

In fondo al listato c'è la subroutine di calcolo per i numeri a 5 byte: se il primo byte è 0, allora il numero è uno "small integer" e viene eseguita la linea 9936, altrimenti il calcolo è fatto alla linea 9938.

Note applicative

Ed ora un trucchetto molto utile per liberare lo spazio (linee 9878 - 9938) occupato dalla routine: occorre rilocare la routine sopra la zona delle linee editabili, cioè dalla linea 10000 in sù.

Come fare? Innanzitutto occorre modificare i salti così:

linea	salto a
9810	10013
9812	10018
9814	10025
9816	10029
9818	10038
9820	10046
9822	10058
9828	10017
9832	10001
9848	10005
9880	10042
9882	10039

poi: linee 9848, 9856, 9874, 9890, 9914, 9910 salto a 10005;

linee 9852, 9866, 9886, 9918 salto a 10068;

linee 9860, 9894 salto a 10062 e cancellare la linea "9878 STOP".

Quindi occorre rinumerare le linee della routine con base 10000 ed incremento 1. Questa operazione può essere realizzata automaticamente sfruttando il renumber pubblicato su **Personal Software** n. 15 a pag. 70.

Se avete un renumber che non funziona fuori dal range delle linee editabili, rinumerate con base 9744 (= 10000 - 256) ed incremento 1; il primo byte del numero di linea conterrà così 38 ed il secondo valori tra 16 e 86. Se non avete alcun renumber eseguite questa rinumerazione manualmente (anzi vi consiglio di eseguirla a matita sulla rivista e di introdurre direttamente il listato rinumerato, ma attenzione a non commettere errori perché non potete fare girare il programma così introdotto). In questi ultimi due casi occorre localizzare manualmente (o mediante un programmino che scriverete appositamente) tutti i primi byte di ogni linea (sapendo che l'indirizzo del primo è dato dalla variabile di sistema PROG (indirizzo 23635/6) e l'indirizzo dei successivi da: P + 4 + PEEK (P + 2) + 256 ★ PEEK (P + 3), dove "P" è l'indirizzo del precedente, e cambiare il valore "38" contenutovi in "39". Così la routine di lista delle variabili è rinumerata a partire dalla linea 10000.

Occorre infine inserire lo stop di protezione alla linea 9999 ("9999 STOP") e salvare la routine. Si noti che il numero di linea 10000 non può

Lista delle variabili per Spectrum

```
9798 STOP
9800 CLOSE #2: CLS
9802 PRINT '"** LISTA DELLE VARI
ABILI **"''
9804 DEF FN h(z)=PEEK z+256*PEEK
 (z+1)
9806 LET v=FN h(23627)
9808 LET endx=FN h(23641)
9810 IF v>=endx OR PEEK v=128 TH
EN GO TO 9826
9812:  IF PEEK v>= 64 AND PEEK v<
= 95 THEN GO TO 9836
9814:  IF PEEK v>= 96 AND PEEK v<
=127 THEN GO TO 9850
9816:  IF PEEK v> 128 AND PEEK v<
=159 THEN GO TO 9858
9818:  IF PEEK v>=160 AND PEEK v<
=191 THEN GO TO 9876
9820:  IF PEEK v>=192 AND PEEK v<
=223 THEN GO TO 9892
9822:  IF PEEK v>=224 AND PEEK v<
=255 THEN GO TO 9916
9824 PRINT FLASH 1;"Esecuzione n
on corretta !": STOP
9826 PRINT '"Variabili usate dal
la routine :  'v','endx','z','z9
','w'."'
9828 IF PEEK 23578<>6 THEN GO TO
 9834
9830 PRINT #1;"Vuoi la stampa ?
(s/n)": PAUSE 0: INPUT ""
9832 IF INKEY$="s" THEN OPEN #2,
"p": CLS : GO TO 9802
9834 CLOSE #2: STOP
9836 PRINT "STRING  VAR:";TAB 12
;CHR$ (PEEK v+32);"$ = '";
9838:  FOR z=v+3 TO v+3-1+FN h(v+
1)
9840:     PRINT CHR$ PEEK z;
9842:  NEXT z
9844:  PRINT "'"
9846:  LET v=v+3+FN h(v+1)
9848:  GO TO 9810
9850 PRINT "NUMERIC VAR:";TAB 12
;CHR$ PEEK v;" = ";
9852:  GO SUB 9936
9854:  LET v=v+6
9856:  GO TO 9810
9858 PRINT "NUMER.ARRAY:";TAB 12
;CHR$ (PEEK v-32);"(";
9860:  GO SUB 9924
9862:  LET z=v
9864:  FOR v=z+3+2*PEEK (z+3) TO
z+1+FN h(z+1) STEP 5
9866:     PRINT TAB 6-LEN STR$ z9;
z9;" = ";: GO SUB 9936
9868:     LET z9=z9+1
9870:  NEXT v
9872:  LET v=z+3+FN h(z+1)
9874:  GO TO 9810
9876 PRINT "NUMERIC VAR:";TAB 12
;CHR$ (PEEK v-64);
9878:  LET v=v+1
9880:  IF PEEK v>=128 THEN GO TO
9884
9882:  PRINT CHR$ PEEK v;: GO TO
9878
9884:  PRINT CHR$ (PEEK v-128);"
= ";
9886:  GO SUB 9936
9888:  LET v=v+6
9890:  GO TO 9810
9892 PRINT "CHAR. ARRAY:";TAB 12
;CHR$ (PEEK v-96);"$(";
9894:  GO SUB 9924
9896:  FOR z=v+4+2*PEEK (v+3) TO
v+2+FN h(v+1) STEP FN h(v+2+2*PE
EK (v+3))
9898:     PRINT TAB 6-LEN STR$ z9;
z9;" = '";
9900:     FOR w=z TO z-1+FN h(v+2+
2*PEEK (v+3))
9902:        PRINT CHR$ PEEK w;
9904:     NEXT w
9906:     PRINT "'"
9908:     LET z9=z9+1
9910:  NEXT z
9912:  LET v=v+FN h(v+1)+3
9914:  GO TO 9810
9916 PRINT "FORNEXT VAR:";TAB 12
;CHR$ (PEEK v-128);" = ";
9918:  GO SUB 9936
9920:  LET v=v+19
9922:  GO TO 9810
9924 FOR z=v+4 TO v+2+2*PEEK (v+
3) STEP 2
9926:  PRINT FN h(z);"," AND z<v+
2+2*PEEK (v+3);
9928 NEXT z
9930 PRINT ")"
9932 LET z9=1
9934 RETURN
9936 IF PEEK (v+1)=0 THEN PRINT
FN h(v+3)-(65536 AND (PEEK (v+2)
=255)): RETURN
9938 PRINT ((PEEK (v+2)<128)-(PE
EK (v+2)>=128))*2↑(PEEK (v+1)-16
0)*(2↑31+2↑24*(PEEK (v+2)-128*(P
EEK (v+2)>=128))+2↑16*PEEK (v+3)
+2↑8*PEEK (v+4)+PEEK (v+5)): RET
URN
```

Listato 1. *Listato della routine, scritta interamente in BASIC. Potete introdurre la routine com'è ed utilizzarla con GOTO 9800, oppure rilocarla sopra la zona delle linee editabili, secondo la procedura descritta nell'articolo, e utilizzarla con GOTO 10000.*

```
** LISTA DELLE VARIABILI **

FORNEXT VAR:i = 11
NUMERIC VAR:pippo = 32768
NUMER.ARRAY:a(3)
     1 = 10
     2 = 255
     3 = 65535
CHAR. ARRAY:z$(2)
     1 = 'ab'
FORNEXT VAR:v = 26293
NUMERIC VAR:endx = 26403
FORNEXT VAR:z = 26293
NUMERIC VAR:z9 = 2
FORNEXT VAR:w = 26293
STRING  VAR:a$ = 'pluto'
CHAR. ARRAY:d$(2,10)
     1 = 'primo     '
     2 = 'secondo   '

Variabili usate dalla routine :
 'v','endx','z','z9','w'.
```

Figura 1. *Risultato della routine presentata con alcune variabili di prova. Si noti che viene indicato anche il tipo di variabile.*

apparire nel listato ma verrà visualizzato come ":000". Per usare la routine occorre ovviamente eseguire GOTO 10000.

Ultimo trucchetto: per memorizzare il programma date CLEAR per eliminare le variabili di prova e scrivete:

SAVE "VARIABLES" + CHR$ 240

che permette un elegante nome di 14 caratteri. ■

PERSONAL SOFTWARE

Loquax per Spectrum

Volete far parlare il vostro computer? Rimboccatevi le maniche e copiatevi questo programma ...

di *Cesare Giavani*

SINTETIZZATORE		DIGITALIZZATORE	
32806/7	Inizio area dati.	**32770/1**	Inizio area dati.
32809/10	Numero byte.	**32773/4**	Numero byte.
32815	Durata interciclo di ritardo.	**32779**	Durata interciclo di ritardo.

N.B. - Modifiche alla durata dei cicli di ritardo: più si abbassa il valore più aumenta la precisione e l'occupazione di memoria.
Più lo si aumenta più decresce la precisione e l'occupazione di memoria dei dati-voce.

Figura 1. *Indirizzi dei dati interni alle due routine che possono essere modificati da BASIC per ottenere l'effetto voluto.*

Per sapere come utilizzare efficacemente Loquax occorre spendere qualche parola sulla teoria della sintesi vocale. Il computer, per poter riprodurre la voce, ha bisogno di dati che, utilizzati per pilotare l'altoparlante, riproducano determinati suoni. Dunque per prima cosa serve una routine in linguaggio macchina che trasformi degli impulsi di origine sonora in dati discreti da porre poi in un'area di memoria stabilita; questa routine è il "Digitalizzatore".
Si vuole poi che il computer esegua l'operazione inversa, cioè che trasformi numeri in suoni; per questo serve una buona routine in linguaggio macchina, il cosiddetto "Sintetizzatore", (per gli assembleromani allego un disassemblato e i flowchart delle routine in questione).

Ora scendiamo nel pratico visto che nulla serve più del toccare con mano: per fare sì che il tuo Spectrum dica "Salve amico" devi eseguire le operazioni seguenti.
1) Prendi un nastro pulito e, con parole chiare e ben scandite, registra la tua voce che dice: "Salve amico". Completata la registrazione, riavvolgi il nastro e posizionalo subito prima delle parole appena incise.
2) Lancia Loquax e rispondi con l'opzione 1 al menu.
Inserisci ora l'indirizzo di partenza dell'area dati (32845) poi, come numero dei byte da utilizzare, inserisci 8000, od un altro numero scelto, tenendo presente che un secondo richiede circa 1000 byte.
Fai partire il registratore e premi subito un tasto.
3) Conclusa la digitalizzazione e fermato il registratore, sei tornato al menu. Ora rispondi con 2. Ti vengono chiesti ancora dei dati, rispondi con gli stessi usati nella fase di digitalizzazione. Premi un tasto, e se il volume del registratore e se la registrazione erano entrambi buoni, sentirai il computer dire: "Salve amico". Ricorda che è necessario usare un livello piuttosto alto.

Se tutto è andato bene certamente sarai ansioso di utilizzare Loquax per dare ai tuoi programmi effetti vocali. Per fare ciò prepara una registrazione contenente una di seguito all'altra tutte le parole che intendi utilizzare nel tuo gioco e, dando un numero di byte piuttosto alto circa 30.000, procedi come prima alla digitalizzazione.
Fatto ciò riascolta l'insieme delle parole e trova per tentativi i valori di partenza e di lunghezza che consentono di sentire una parola alla volta, ed annotali.
Tali valori inseriti nelle opportune posizioni del sintetizzatore, causeranno, al momento della chiamata randomize USR 32804 l'emissione del beeper delle parole volute: per esempio "Game over" a fine gioco, oppure "Allert invasion" mentre stai per essere sopraffatto da orde di alieni.
Adesso, ti consiglio di salvare i dati in memoria con l'opzione 3 dando come indirizzo di partenza 32804 in modo da salvare anche il Sintetizzatore.
Dopo aver verificato i dati puoi dare il NEW e caricare il tuo programma che però, N.B., non può occupare più di 16 Kbyte, dato che ho riservato 32 Kbyte di memoria ai dati vocali, come noto molto ingombranti.
Comunque, chi volesse più spazio per il suo programma, può rilocare convenientemente le due routine in linguaggio macchina, dato che esse non contengono salti assoluti.
Va solo curato che il Digitalizzatore non memorizzi dati-voce sopra il Sintetizzatore.
A questo punto la cosa migliore è sperimentare tutte le idee per scoprire tutto ciò che Loquax può fare.

Per concludere qualche cenno sulle routine in linguaggio macchina. Queste si presentano in forma molto spartana, che è il miglior compromesso tra efficienza e occupazione di memoria; comunque sono facilmente migliorabili da chiunque abbia dimestichezza con l'Assembly. ■

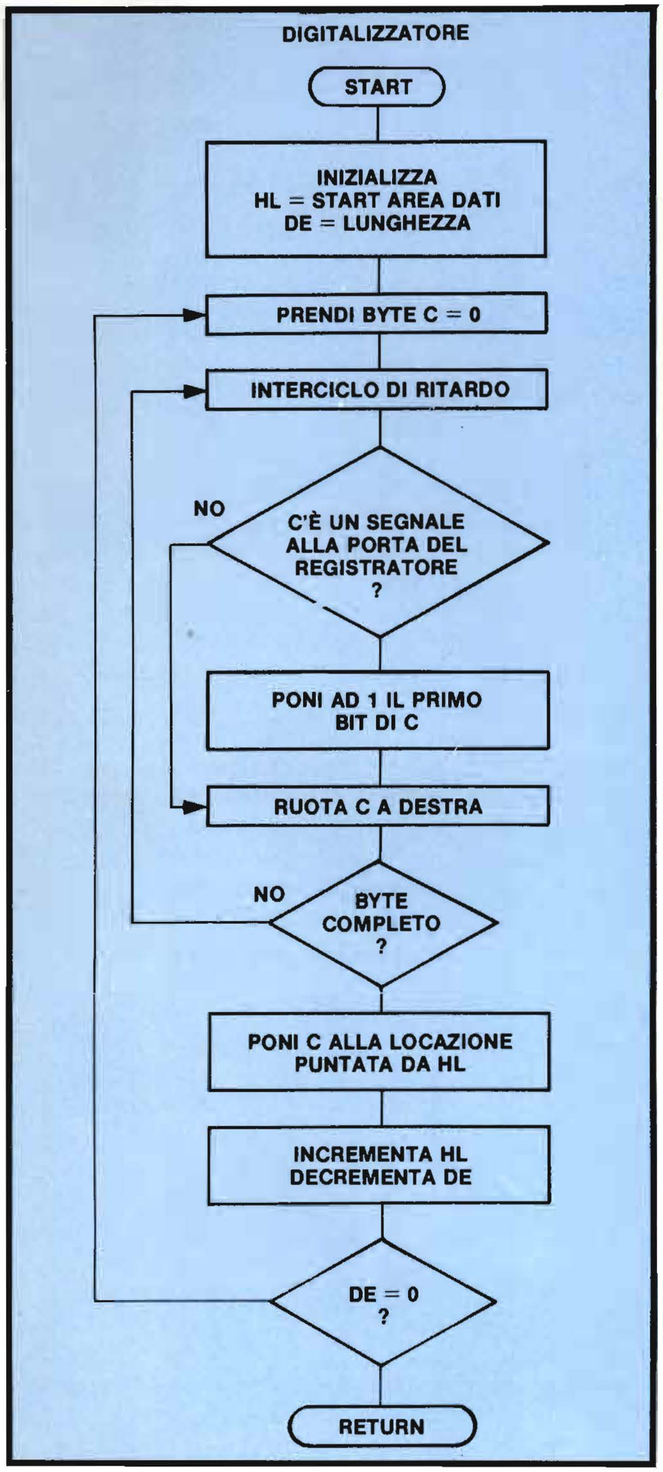


Figura 2. *Diagramma di flusso della routine di digitalizzazione.*

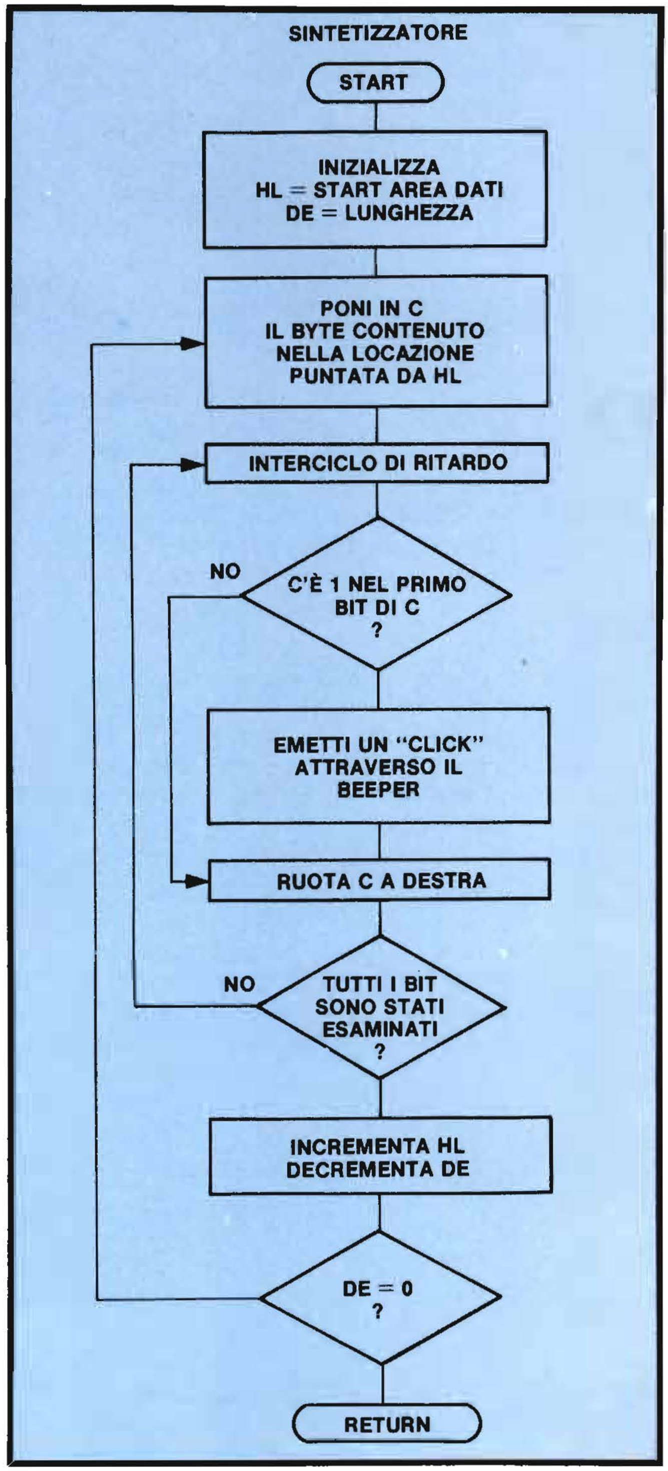


Figura 3. *Diagramma di flusso della routine di riconversione in suono.*

```
   1 REM ##########################
         #                        #
         #         LOQUAX         #
         #                        #
         #        © 1984          #
         #                        #
         #     Giavani Cesare     #
         ##########################
   2 REM  programma di sintesi
   3 REM  vocale per Spectrum.
   4 REM  necessita di 48K RAM
   5 REM
   6 BORDER 1: PAPER 1: INK 7: C
LS : CLEAR 32767
  10 DEF FN k(X)=x-INT (x/256)*2
56
  11 DEF FN l(x)=INT (x/256)
  12 DEF FN j(x)=PEEK x+256*PEEK
 (x+1)
  20 GO SUB 1000
  30 CLS : BEEP .1,30
  32 PRINT TAB 14;"LOQUAX"
  33 PRINT 'TAB 6;"© 1984 Giavan
i Cesare"
  34 PRINT ''"Menu:";TAB 10;"■ D
igitalizza"''
  35 PRINT TAB 10;"■ Sintetizza"
''
  36 PRINT TAB 10;"■ Salva dati"
''
  37 PRINT TAB 10;"■ Carica dati
"''
  38 PRINT TAB 10;"Default  Menu
"
  40 INPUT r: IF r<1 OR r>4 THEN
 GO TO 30
  43 CLS
  45 GO SUB 100*r
  46 CLS
  50 GO TO 30
 100 PRINT TAB 10;"DIGITALIZZATO
RE"'': BEEP .2,1
 105 PRINT "Start area dati:3284
5"''
 107 INPUT "da che indirizzo par
to? ";s
 108 IF s<32845 THEN GO TO 107
 110 PRINT "start:";s''
 113 INPUT "Quanti bytes utilizz
i? ";d
 115 PRINT "Lunghezza:";d''
 120 BEEP .5,20
 130 PRINT "Dai il PLAY al regis
tratore"
 135 PRINT AT 21,0;"premi un tas
to per digitalizzare"
 140 PAUSE 0
 141 BEEP .02,40
 142 PRINT AT 21,0;"
                    ";AT 5,0;
 145 POKE 32770,FN k(s): POKE 32
771,FN l(s): POKE 32773,FN k(d):
 POKE 32774,FN l(d)
 150 RANDOMIZE USR 32768
 160 RETURN
 200 PRINT TAB 10;"SINTETIZZATOR
E"'': BEEP .2,2
 205 PRINT "Start area dati:3284
5"''
 210 INPUT "Da che indirizzo par
to? ";s
 220 INPUT "Quanti bytes? ";d
 230 PRINT AT 21,0;"premi un tas
to per ascoltare"
 232 PAUSE 0
 233 BEEP .03,40: PRINT AT 21,0;
"
 ";AT 5,0;
 235 POKE 32806,FN k(s): POKE 32
807,FN l(s): POKE 32809,FN k(d):
 POKE 32810,FN l(d)
 240 RANDOMIZE USR 32804
 250 INPUT "ti soddisfa? (s/n)";
r$
 255 IF r$(1)="s" THEN PRINT "AN
NOTA start:";s;" bytes:";d
 257 INPUT "vuoi tornare al menu
? (s/n)";r$
 260 IF r$(1)="s" THEN RETURN
 270 CLS : GO TO 200
 300 PRINT TAB 10;"SAVE dei DATI
"'': BEEP .2,3
 310 PRINT "Start area dati:3284
5"''
 320 INPUT "da dove vuoi salvare
? ";s
 325 INPUT "quanti bytes? ";d
 330 INPUT "nome del file?";r$
 340 SAVE r$CODE s,d
 350 RETURN
 400 PRINT TAB 10;"LOAD di DATI"
'': BEEP .2,4
 410 PRINT "Start area dati:3284
5"''
 420 INPUT "nome del file? ";r$
 430 LOAD r$CODE
 450 RETURN
 600 REM ***********************
 610 REM * linguaggio macchina *
 620 REM ***********************
1000 REM   digitiz & sintetiz
1005 POKE 23609,20
1010 FOR n=32768 TO 32844
1020 READ a: POKE n,a
1030 NEXT n
1035 RETURN
1050 DATA 243,33,0,0,17,0,0,1,0,
8,62,20,61,32,253,219,254,230,64
,40,2,203,193,203,9,16,239,113,3
5,27,122,179,32,229,251,201
1060 DATA 243,33,0,0,17,0,0,78,6
,8,62,19,61,32,253,58,72,92,15,1
5,15,203,65,40,2,238,56,211,254,
203,9,16,233,35,27,122,179,32,22
4,251,201
9999 SAVE "LOQUAX" LINE 1
```

Listato 1. *Master BASIC utile per effettuare la digitalizzazione del suono e determinare i valori necessari per il corretto riascolto mediante il sintetizzatore. Le routine in linguaggio macchina sono contenute in linee "DATA" alla fine del listato.*

```
ESEMPIO USO DEL SINTETIZZATORE

   5 CLEAR 32803: PRINT AT 10,5;
 PAPER 2; INK 7; FLASH 1;"CARICA
MENTO LOQUAX CODE ": PRINT AT 20
,6;"PREGO ATTENDERE..."
  10 LOAD ""CODE
  20 BORDER 6: PAPER 6: INK 9: C
LS
  30 PRINT TAB 10; PAPER 1; INK
7; FLASH 1;"LOQUAX DEMO"
  35 PRINT
  50 DEF FN k(x)=x-INT (x/256)*2
56
  60 DEF FN l(x)=INT (x/256)
  80 DIM s(6): DIM d(6)
  85 PRINT "allert",1
  90 LET s(1)=33845: LET d(1)=11
00
  95 PRINT "invasion",2
 100 LET s(2)=35830: LET d(2)=10
00
 105 PRINT "game over",3
 110 LET s(3)=37000: LET d(3)=14
00
 115 PRINT "player one",4
 120 LET s(4)=39000: LET d(4)=10
00
 125 PRINT "get ready",5
 130 LET s(5)=40990: LET d(5)=14
00
 135 PRINT "player two",6
 140 LET s(6)=42200: LET d(6)=13
00
 150 PRINT AT 20,6;"© 1984 Giava
ni Cesare"
 155 PRINT #1;"premi 1 2 3 4 5 6
 per scegliere"
 160 LET a$=INKEY$
 165 LET a=CODE a$-48
 170 IF a<1 OR a>6 THEN GO TO 16
0
 190 POKE 32806,FN k(s(a)): POKE
 32807,FN l(s(a))
 200 POKE 32809,FN k(d(a)): POKE
 32810,FN l(d(a))
 210 RANDOMIZE USR 32804
 220 GO TO 160
```

Listato 2. *Programma esemplificativo per generare parole in seguito alla pressione di un tasto. Non è di alcuna utilità inserire questo programma senza aver memorizzato una serie di parole e determinati i valori per il sintetizzatore (vedi procedura descritta nell'articolo).*
L'indirizzo di partenza e la lunghezza di ogni parola sono contenuti rispettivamente nei vettori "s()" e "d()".

```
DIGITALIZZATORE
DISASSEMBLATO

8000 F3       DI
8001 214D80   LD     HL,804D
8004 11401F   LD     DE,1F40
8007 010008   LD     BC,0800
800A 3E14     LD     A,14
800C 3D       DEC    A
800D 20FD     JR     NZ,800C
800F DBFE     IN     A,(FE)
8011 E640     AND    40
8013 2802     JR     Z,8017
8015 CBC1     SET    0,C
8017 CB09     RRC    C
8019 10EF     DJNZ   800A
801B 71       LD     (HL),C
801C 23       INC    HL
801D 1B       DEC    DE
801E 7A       LD     A,D
801F B3       OR     E
8020 20E5     JR     NZ,8007
8022 FB       EI
8023 C9       RET
```

Listato 3. *Disassemblato della routine di digitalizzazione.*
Tutti i valori numerici sono in esadecimale. Si rammenta che le quattro colonne contengono in ordine: indirizzo, codifica numerica, mnemonico dell'istruzione e operandi, secondo il formato Assembly standard.

```
SINTETIZZATORE
DISASSEMBLATO

8024 F3       DI
8025 214D80   LD     HL,804D
8028 11401F   LD     DE,1F40
802B 4E       LD     C,(HL)
802C 0608     LD     B,08
802E 3E13     LD     A,13
8030 3D       DEC    A
8031 20FD     JR     NZ,8030
8033 3A485C   LD     A,(5C48)
8036 0F       RRCA
8037 0F       RRCA
8038 0F       RRCA
8039 CB41     BIT    0,C
803B 2802     JR     Z,803F
803D EE38     XOR    38
803F D3FE     OUT    (FE),A
8041 CB09     RRC    C
8043 10E9     DJNZ   802E
8045 23       INC    HL
8046 1B       DEC    DE
8047 7A       LD     A,D
8048 B3       OR     E
8049 20E0     JR     NZ,802B
804B FB       EI
804C C9       RET
```

Listato 4. *Disassemblato della routine di riconversione sonora. Tutti i valori numerici sono in esadecimale.*

Nimble per Spectrum

Un articolo riservato ai nostri lettori "ciberneticamente" più avanzati

di *P. Francesco Ghedini*

Macchine e pensiero

"Che cosa significa AM?
Gli rispose Gorrister. Avevamo ripetuto quella sequenza già mille volte, ma per Benny era sempre una novità.
All'inizio significava Allied Mastercomputer, e poi Adaptative Manipulator e più tardi divenne senziente e si collegò, e allora lo chiamarono Agressive Menace; ma ormai era troppo tardi e alla fine si diede il nome di AM, intelligenza emergente, e intendeva dire - Io sono - ... cogito, ergo sum... penso, dunque sono.
Benny sbavò un poco, e ridacchiò".
Tratto da "Non ho bocca e devo urlare"
di *Harlan Ellison*

Quale immagine più calzante, più simbolicamente significativa, di un immane perverso calcolatore pensante, poteva trovare Ellison per dare corpo alla figura del male nel suo magistrale racconto?
Forse nessuna! Il calcolatore nella mitologia dell'uomo del ventesimo secolo è allegoria della Tecnica o, se vogliamo, più in generale del sapere scientifico. Un sapere che per l'uomo della strada diventa sempre più difficile da comprendere proprio perché sempre più specialistico, per iniziati, non dominabile e quindi pauroso.

Ma a parte questo, è razionalmente proponibile l'idea di un calcolatore pensante? O in altre parole possiamo chiederci: "Possiamo costruire, o perlomeno progettare, una macchina pensante?".
Alla luce delle attuali conoscenze siamo ben lontani dal poter dare una risposta a questa affascinante domanda, ciononostante possiamo fare alcune importanti considerazioni al riguardo.
Sarà utile, per introdurre l'argomento, ricorrere ad un programma che pur giocando un gioco forse banale, possiede caratteristiche per alcuni versi sorprendenti.

Un gioco per cominciare

Sicuramente molti conosceranno il gioco del Nim (vedi riquadro 1). Il programma che vi propongo simula un giocatore di Nim, che affina le proprie abilità di gioco con l'aumentare del numero di partite disputate (vedi riquadro 2).
Per poter realizzare ciò praticamente, mi sono ampiamente servito della descrizione delle "macchine a scatoline" riportata al capitolo n. 8 di "Giochi ed enigmi matematici Vol. 4", curato da Martin Gardner, libro al quale rimando chiunque interessato all'argomento per maggiori notizie.

Il Nim è un gioco che gode di vasta popolarità e come tale si può giocare in molti diversi modi che pur non cambiano sostanzialmente lo spirito del gioco.
Nella versione più nota si parte con un numero arbitrario di stecchini disposti su tre righe diverse (generalmente più di tre stecchini per riga altrimenti il gioco diventa banale); a turno si tolgono uno o più stecchini, a patto che stiano sulla stessa riga. Vince chi riesce a togliere l'ultimo stecchino.
Il Prof. Bouton dell'Università di Harvard scoprì che esiste il modo di vincere sempre per il primo giocatore anche nel caso molto generale di n pezzi su n righe. Per fare ciò bisogna scrivere il numero di pezzi di una riga in base due quindi incolonnando i numeri si devono sommare le cifre colonna per colonna: se le somme sono zero o pari la posizione è sicura altrimenti la posizione è pericolosa.
Questo significa che se il primo giocatore gioca al meglio farà solo mosse che non permettono al secondo di togliere l'ultimo stecchino assicurandosi in questo modo la vittoria.
Per quello che riguarda il programma ho voluto adottare una piccola variante alle regole del gioco: per vincere occorre lasciare all'avversario l'ultimo stecchino o in altre parole perde chi prende l'ultimo. Questo rende inattuabile la strategia prima descritta e costringe il giocatore umano a cercare una nuova strategia.

I	001
II	010
III	011 posizione sicura secondo la strategia prima descritta
	022

II	010
III	011 posizione insicura
	021

Riquadro 1. *Spiegazione del gioco del Nim.*

Nimble per Spectrum

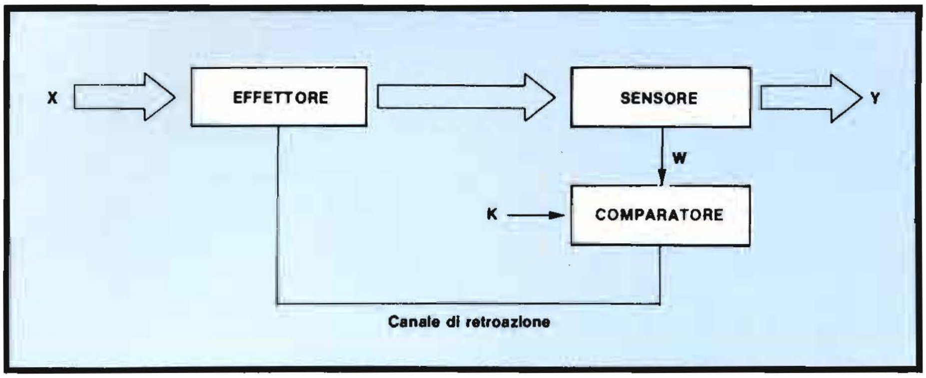


Figura 1. *Schema logico di un sistema a retroazione.*
X rappresenta l'entrata del sistema, Y rappresenta l'uscita del sistema, W (contenuto del sistema) è funzione di X. Il sensore misura W e immette il dato in un comparatore ove esso è confrontato con il valore di riferimento K; se vi è differenza il comparatore-attraverso il canale di retroazione - modifica X in modo da annullare lo scostamento.

Il meccanismo fondamentale che sta alla base del programma, è molto semplice e cercherò di darne una breve descrizione.
Il programma inizialmente non possiede alcun tipo di strategia, ma ha immagazzinate, come dati iniziali, tutte le possibili mosse legali (cioè l'elenco delle mosse possibili in ogni possibile situazione). Ogni volta che il programma sbaglia e perde la partita, elimina dall'elenco delle mosse possibili quella che ha determinato la sua sconfitta. Così facendo il programma non commetterà mai due volte lo stesso errore (vedi figura 7).
Il programma può giocare sia per primo che per secondo e, cosa assai importante, può aumentare la propria abilità giocando contro se stesso.
Questo programma che ho voluto chiamare Nimble, come l'analoga macchina a scatoline di Stuart C. Hight citata nel volume di Gardner, ha caratteristiche estremamente interessanti che sarà bene analizzare in dettaglio.
Prima di tutto possiamo notare che Nimble ha quello che si dice un comportamento *teleologico*, cioè tende a raggiungere una meta.
Possiamo infatti distinguere uno stato iniziale (in cui Nimble è assolutamente privo di strategia e compie una qualsiasi delle mosse legali in una data situazione), uno stato finale (in cui Nimble ha eliminato tutte le mosse che portano alla sconfitta; ovviamente se non esiste mai la possibilità di vincere, per il programma, saranno eliminate tutte le mosse dalla sua memoria per cui si rifiuterà sistematicamente di giocare) e possiamo dire che Nimble tenderà sistematicamente a raggiungere lo stato finale.
Perché tutto ciò sia possibile ho dovuto dotare il programma di tre strutture fondamentali:
1) degli organi effettori interni, cioè degli organi in grado di modificare lo stato;
2) di una struttura che mette in grado il sistema di rilevare la differenza fra lo stato attuale e lo stato perseguito;
3) di una struttura che metta in grado il sistema di tradurre l'informazione rilevata, in comandi atti a ridurre la differenza rilevata.
In altre parole ho dovuto implementare un tipico circuito a retroazione (vedi figura 1).
L'effetto macroscopicamente evidente del comportamento teleologico di Nimble è l'affinamento delle sue capacità di gioco. In altre parole la struttura che è alla base del funzionamento del programma mette in grado il programma stesso di modificare il proprio comportamento in funzione di determinati stimoli, ma allora possiamo dire che Nimble è in grado di "apprendere" (seppure nella limitatissima gamma di situazioni previste dal programma). Se infatti richiamiamo la definizione di apprendimento che danno W.N. Kellog e S.H. Britt ("l'apprendimento è un mutamento o una modifica permanente del comportamento risultante dall'attività dell'organismo stesso e tendente ad adattare l'organismo all'ambiente") noi vediamo che essa si adatta quasi perfettamente anche al programma a patto che, o generalizziamo il concetto di organismo, o lo sostituiamo con il termine più generale di apparato (vedi figura 3).
Ma non è tutto, infatti se torniamo ancora una volta ad analizzare il processo di apprendimento del programma, noi notiamo che in definitiva esso si identifica col passaggio da uno stato contenente n mosse ad uno stato contenente (n-1) mosse (vengono eliminate una per volta le mosse che portano alla sconfitta).
Possiamo quindi, generalizzando, osservare che il processo di adattamento all'ambiente consiste nella banalizzazione di determinati processi di tipo superiore (come la scelta casuale fra più possibili mosse diverse) in semplici processi del tipo causa-effetto (il che avviene quando rimane un'unica mossa possibile o

quando non ne esiste alcuna e compare la scritta "HO PERSO") (vedi figura 2).

Mi si permetta a questo punto di ricordare una frase dello psicologo R.J. Angell secondo il quale psiche e corpo sono "poli funzionali della vita dell'organismo, così che quest'ultimo avendo come compito precipuo l'adattamento di sé all'ambiente - fisico e sociale - in cui si trova, utilizza i processi coscienti al punto in cui nuove coordinazioni senso-motorie sono state stabilite, le quali, in seguito perfezionandosi, permettono una graduale diminuzione dell'aspetto mentale (cosciente) del processo, finché, nel caso limite, la coscienza interamente scompare e la coordinazione diviene completamente automatica".

Vorrei chiarire meglio quello che intendo dire: non sto assolutamente dicendo che a mio parere la coscienza può essere ricondotta ad una scelta casuale fra determinate alternative, voglio solo mettere in evidenza l'analogia formale che esiste fra il processo adattativo del programma descritto prima, e ciò che avviene nell'uomo; il paragone non intende andare al di là di tanto.

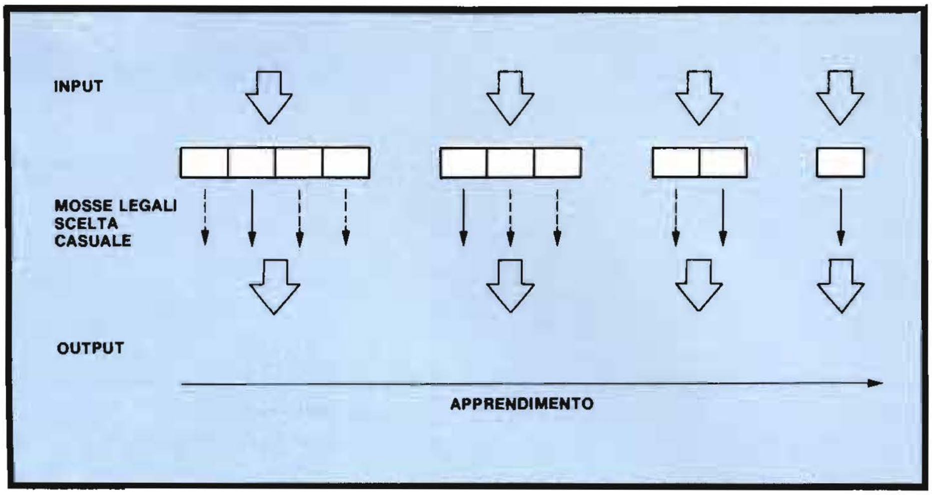


Figura 2. *Processo di apprendimento: stati successivi del programma.*

Le tesi funzionalistiche

Nella prima parte sono state messe in evidenza alcune analogie strutturali fra Nimble e alcuni aspetti del funzionamento del cervello ma, vediamo ora di chiarire quali sono i

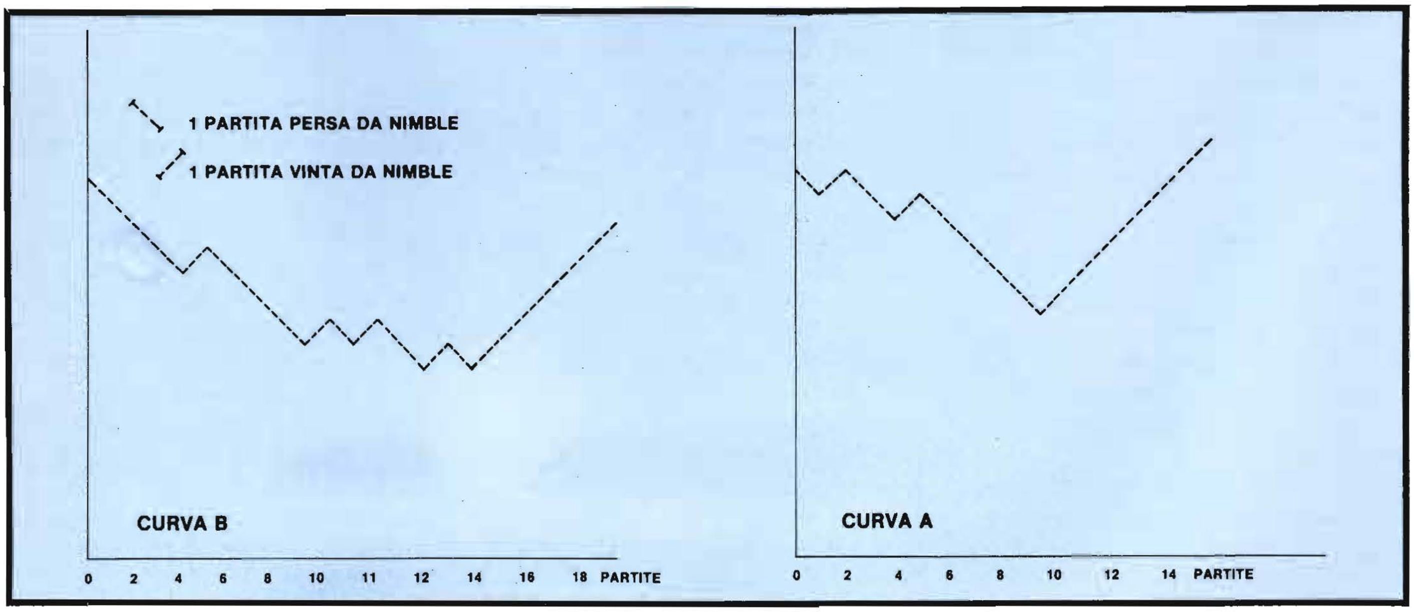


Figura 3. *Processo di apprendimento: Curve di apprendimento di Nimble nelle prime 18 partite. Nella curva A è descritto il comportamento di Nimble quando il giocatore umano muove per secondo e gioca sempre al meglio. Nella curva B il giocatore umano muove per primo, effettuando una mossa casuale fra le sole tre possibili, quindi gioca al meglio.*

postulati teorici che rendono un tale accostamento proponibile.
L'attività mentale può essere spiegata, fondamentalmente, alla luce di tre punti di vista:
1) il punto di vista strutturale (analitico anatomico) che mira a trarre in luce gli elementi della vita mentale, o a chiarire la natura dei meccanismi del comportamento. Si chiederà perciò quale sia il sostrato del pensiero, quale sia la struttura della emozione, quali siano i sentimenti elementari;
2) il punto di vista meccanico tecnico (analitico operativo), cioè la spiegazione dei rapporti di inserimento fra i vari meccanismi che l'analisi strutturale aveva rivelato, per esempio come agisce il sentimento sulle operazioni dell'intelligenza;
3) il punto di vista funzionale (dinamico relazionale) vuole spiegare la funzione di un particolare processo nella vita dell'individuo. Intende investigare a quali bisogni rispondano determinati fenomeni, in che senso sono dei mezzi di adattamento, insomma che relazione hanno con la condotta totale dell'individuo.
Il primo punto di vista risponde alla domanda "quid", il secondo alla domanda "quomodo", il terzo alla domanda "cur".
Il terzo punto di vista ha almeno due innegabili vantaggi:
1) non si preoccupa di definire dogmaticamente la natura ultima e la relazione intercedente fra psiche e materia;
2) si limita ad indicare il vero rapporto vitale che caratterizza il dinamismo organico e che la semplice osservazione empirica è in grado di verificare.
Figlio del terzo modo di intendere l'attività mentale è il moderno funzionalismo (altrimenti detto Cognitivismo e naturale evoluzione del

1) Caricare il programma.
2) Dare il RUN.
Dopo la scritta di presentazione il programma chiede: "Vuoi che giochi contro me stesso per migliorare la mia strategia?" se si intende fare allenare il programma prima di sfidarlo rispondere con una "s", altrimenti rispondere con qualsiasi altro tasto.
In caso di risposta affermativa il programma chiede il numero di partite per il quale si deve allenare: un grado di apprendimento medio si ottiene con dieci partite.
3) Compare la situazione iniziale di gioco.
Se si intende fare giocare per primo il programma, rispondere "0" alla domanda:
"RIGA?"
in caso contrario inserire il numero della riga dalla quale si intende togliere i pezzi.
A questo punto il programma chiederà quanti
"PEZZI?"
togliere dalla riga scelta.
4) Il programma mostra la situazione dopo la vostra mossa e, dopo pochi secondi, la situazione dopo la sua mossa.
5) Ripetere il passo 3 fino a che non verrà visualizzato il messaggio
"HO VINTO"
oppure il messaggio
"HO PERSO".
6) Se si intende giocare un'altra partita premere "s", altrimenti qualsiasi altra lettera.
Il programma automaticamente risistema, dopo ogni mossa del giocatore, i pezzi rimasti in modo che la riga numero 1 abbia un numero di pezzi maggiore o uguale al numero di pezzi della riga numero 2 e che la riga numero 2 abbia un numero di pezzi maggiore o uguale della riga numero 3.
La riga con numero più alto ha sempre il numero minore di pezzi.
Ad esempio la situazione:

I
II
III significa
1 pezzo nella riga numero 3
2 pezzi nella riga numero 2
3 pezzi nella riga numero 1

se ora operiamo la mossa 1-2 cioè due pezzi dalla riga numero 1 il programma visualizzerà la situazione dopo la vostra mossa in questo modo:

I
I
II che significa

1 pezzo nella riga numero 3
1 pezzo nella riga numero 2
2 pezzi nella riga numero 1

Le righe vuote saranno sempre le più alte, e non verranno visualizzate.
Il programma non accetta mosse inesatte e nel caso togliate anche l'ultimo pezzo (perdendo in questo modo la partita in modo non conforme allo spirito del gioco, in quanto scopo del gioco è forzare l'avversario a prendere l'ultimo pezzo) il programma vi avvertirà dell'errore e comincerà una nuova partita.
Il programma gira sullo Spectrum anche non espanso.
Nota: quando date la vostra mossa al computer fate attenzione a non premere inavvertitamente un tasto diverso da una cifra altrimenti il programma si blocca e non è possibile riprendere con CONTINUE, ma bisogna dare GOTO 117.
Si noti anche che durante la fase di apprendimento il computer non visualizza le mosse giocate, ma solo il numero progressivo della partita.

Riquadro 2. *Istruzioni per l'operatore.*

funzionamento storico), i cui assunti teorici danno sicuramente qualche indicazione per cercare di rispondere alla domanda che ci siamo posti all'inizio (riquadro 3). Infatti, il Funzionalismo è la filosofia della mente basata sulla distinzione, messa in luce dalla scienza dei calcolatori, fra hardware e software.

Il funzionalista arriva ad affermare che la psicologia di un sistema, come un essere umano, una macchina o uno spirito disincarnato, non dipende da ciò di cui è composto, ma da come è organizzato ciò di cui è composto.

Il Funzionalismo non esclude la possibilità, per quanto remota, di sistemi meccanici ed eterei che abbiano stati e processi mentali. In linea di principio, secondo queste teorie, tutte le cose diverse da un essere umano potrebbero avere un software umano.

Questi asserti teorici hanno grosse implicazioni pratiche: una tendenza importante delle scienze cognitive è quella di trattare la mente come un dispositivo che manipola simboli. Questo significa che un processo mentale può essere ricondotto ad una operazione su simboli, ma allora un processo mentale può essere simulato con un algoritmo per macchina di Turing e ciò vuol dire che siamo in grado, quanto meno, di progettare una macchina che simuli quel particolare processo mentale.

Il moderno Funzionalismo propone un proprio modello teorico della mente, che può essere o meno accettato in virtù della sua adeguatezza (la discussione di quanto il modello proposto sia adeguato e rispondente ai fatti sperimentali e quali siano invece le sue carenze esula dallo scopo di questo articolo e non ne farò quindi menzione), ma al di là di questo resta da vedere se il modello va inteso come una semplice finzione di tipo euristico o se dobbiamo farne un uso esistenziale.

Funzionalismo: scuola psicologica fondata su principi evoluzionistici e fiorita a cavallo del secolo negli Stati Uniti. Gli psicologi funzionalisti vedono nell'organismo umano l'ultimo stadio del processo evolutivo sia dal punto di vista fisico che psichico, i processi mentali vengono intesi come strategie di cui l'organismo si serve per sopravvivere in un ambiente fisico e biologico il più delle volte ostile. I funzionalisti affrontano in termini specifici le tematiche relative al pensiero e all'intelligenza umana anche se di questi si limitano a sottolineare gli aspetti adattativi e strumentali, mentre successivamente il cognitivismo dimostrerà che è possibile affrontarne lo studio in maniera più ampia.

Cognitivismo: termine con cui si indica uno dei più importanti movimenti della psicologia contemporanea sperimentale, secondo il quale la mente umana funziona come elaboratrice attiva delle informazioni che giungono tramite gli organi sensoriali, in analogia con i servomeccanismi di tipo cibernetico. il cognitivismo al contrario del funzionalismo non costituisce un sistema teoretico altamente organizzato e internamente coerente. Hanno influenzato le teorie cognitiviste la teoria dell'Informazione e la cibernetica che oltre ad un linguaggio inedito nella tradizione psicologica, forniscono un modello dell'organismo umano come sistema complesso in grado di elaborare informazioni, di compiere scelte fra gli elementi in entrata, di porre in atto sui dati selezionati una serie di trasformazioni e un immagazzinamento rapido ed efficace, di raggiungere decisioni dipendenti dai risultati dell'elaborazione compiuta e non predeterminate in partenza dagli stimoli ambientali in entrata.
K Craik fra il 1940-1945 effettua una serie di esperimenti da cui conclude che la mente umana si comporta come un servomeccanismo di tipo cibernetico capace di autocorrezioni ogni 500 millisecondi.
Nel 1960 Miller, Pribram, Galanter pubblicano il volume "Piani e strutture del comportamento" in cui per la prima volta l'analogia di funzionamento fra la mente umana e il computer viene presentata in modo approfondito. In esso si sostituisce il concetto di riflesso con quello di piano di comportamento, che parte dalla premessa secondo cui il soggetto conoscente interagisce con l'ambiente circostante, continuamente verificando la congruenza fra il proprio progetto comportamentale e le condizioni oggettive esistenti. Molti psicologi cognitivisti utilizzano sistematicamente programmi per computer quali modelli capaci di simulare i processi mentali superiori pur essendo consapevoli dei limiti impliciti di questa simulazione. L'informatica e la cibernetica danno grossi contributi alle teorie comportamentali il cognitivismo in particolare mutua da queste termini come "input" e "output" che sostituiscono termini tradizionali come stimolo e risposta.

Riquadro 3. *In poche parole cosa sono il Funzionalismo e il Cognitivismo.*

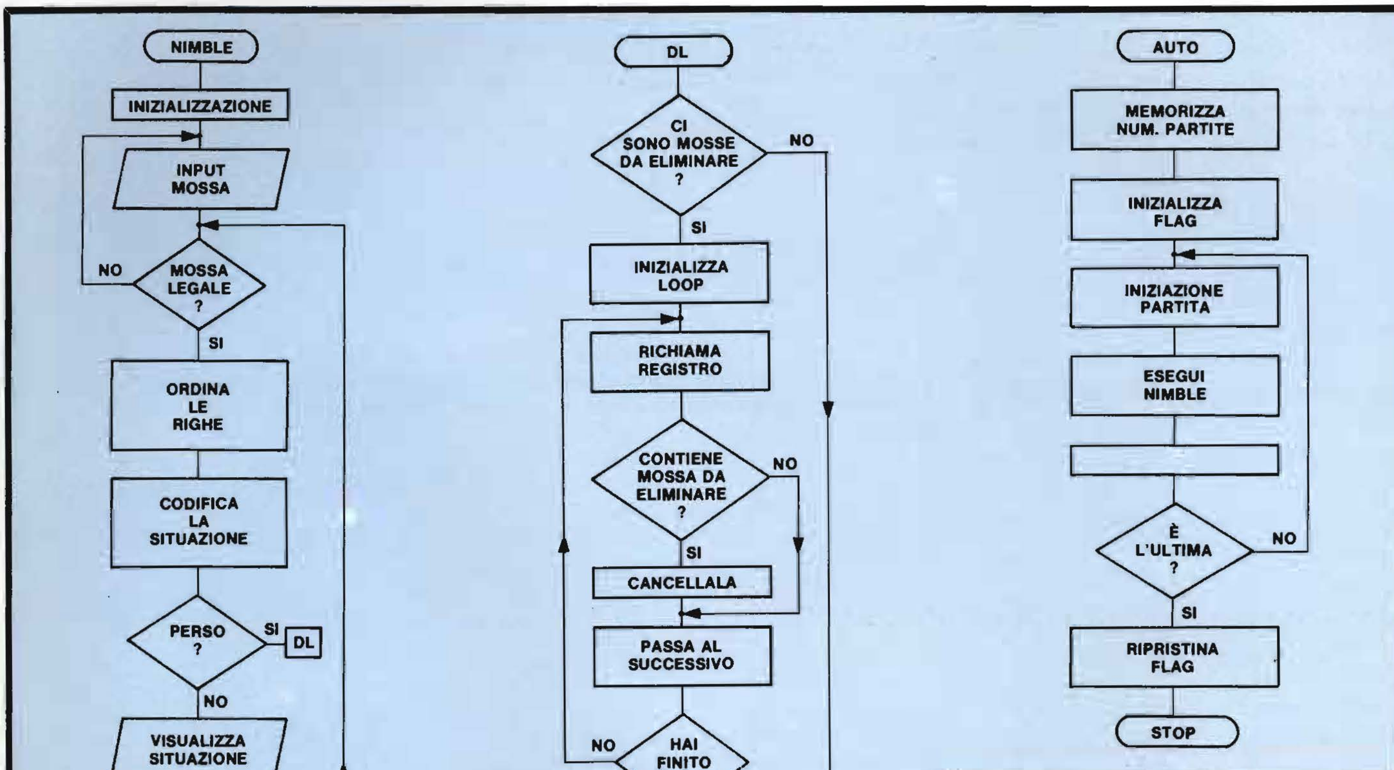
Mi spiego meglio: noi possiamo dire che la mente si comporta *come se fosse* un servomeccanismo di tipo cibernetico, oppure possiamo pensare alla mente *come costituita* da un servomeccanismo di tipo cibernetico.

Purtroppo non siamo ancora in grado di stabilire quale sia la giusta interpretazione, anche se la maggioranza dei ricercatori impegnati in questo campo tende ad usare il modello più come strumento interpretativo che come reale spiegazione della natura ultima della mente.

Conclusioni

La nascita della scienza moderna fu accompagnata da uno sviluppo del pensiero filosofico che portò ad una formulazione estrema del dualismo spirito-materia. René Descartes fondò la propria concezione della Natura su due entità ben distinte: la mente (res cogitans) e la materia (res extensa).

La separazione cartesiana permise



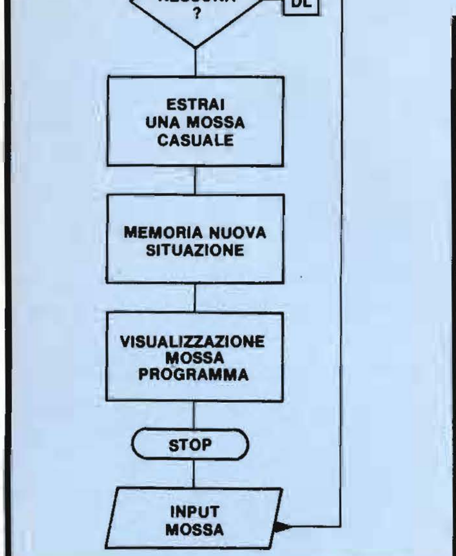



Figura 4. *Diagramma a blocchi di Nimble.*

agli scienziati di considerare la materia come inerte, completamente esterna da se stessi e organizzata in un immane meccanismo.

La filosofia di Cartesio non fu solo importante per lo sviluppo della fisica classica, ma ebbe una enorme influenza sul modo di pensare fino ai giorni nostri. La famosa frase "Cogito, ergo sum" ha portato l'uomo occidentale ad identificarsi con la propria mente, invece che con l'intero organismo. Le più moderne teorie psicologiche tendono a scalzare questo modo di pensare in favore di una maggiore "organicità della mente", tendono cioè a vedere corpo e psiche come una unità funzionale.

Hanno svolto, e svolgono tuttora, un ruolo determinante in questo cambiamento gli studi sul trattamento automatico dell'informazione e lo studio dei meccanismi di controllo, che hanno messo in luce come hardware e software siano solo due facce della stessa medaglia.

Risulta quindi più che mai vero ciò che ebbe occasione di dire il Prof. Silvio Ceccato: "La macchina, l'artefatto, per riuscire a nostra immagine e somiglianza, costringerà a togliere veli e veli alla nostra mente, alla nostra psiche e al nostro soma". E forse è proprio questo l'aspetto più affascinante: capire in misura maggiore chi siamo e dove andiamo. ■

Nimble per Spectrum

Listato 1. *Listato del programma BASIC Nimble. Gira anche su Spectrum 16 Kbyte. Il simbolo della pedina alle linee 2050 e segg. è ottenuto con i caratteri grafici: "ABC" (linea 2050), DEF (linea 2051), GHI (linea 2052).*

```
   1 REM --------------------
         NIMBLE by P.Ghedini,
         revisione P.S.
         --------------------
   2 BORDER 1: PAPER 1: CLS
   3 PRINT AT 10,9; INK 7;"IL MI
O NOME E'": PRINT AT 12,13; INK
7; INVERSE 1;"NIMBLE"
   5 GO SUB 5000: REM definizion
e caratteri
   9 REM immagazzina
         le mosse iniziali
  10 DIM a(3): DIM m$(19,6)
  15 RESTORE 50
  20 FOR i=1 TO 18
  30 READ m$(i)
  40 NEXT i
  50 DATA "abc","bchgd","cgie","
def","eflno","fnqr","or","diekmn
","gdljk","empq","kl","lmn","no"
,"npq","qor","r","q","r"
  60 LET fl1=0: LET fl2=0: INPUT
 PAPER 4; INK 0;"Vuoi che giochi
 contro me stessoper migliorare
la mia strategia (s/n)? ";a$
  70 IF a$="s" THEN GO SUB 4000
  80 BORDER 7
 100 LET a$=""
 110 FOR i=1 TO 3: LET a(i)=3: N
EXT i
 115 GO SUB 2000: PRINT AT 20,1;
 PAPER 4;"Situazione iniziale ":
 PRINT AT 21,1; PAPER 4;" per pa
ssare inserisci '0' "
 117 REM input mossa
 120 PRINT AT 0,0;"Riga ?";
 125 BEEP .2,30: BEEP .1,25: BEE
P .2,30
 130 LET a$=INKEY$
 140 IF a$="" THEN GO TO 130
 150 IF INKEY$<>"" THEN GO TO 15
0
 155 BEEP .12,30
 160 LET a1=VAL a$: PRINT a1
 170 IF a1=0 THEN GO TO 290
 180 IF a1<1 OR a1>3 THEN PRINT
PAPER 4;"errore ": GO TO 120
 185 IF a(a1)=0 THEN PRINT PAPER
 4;"errore ": GO TO 120
 190 PRINT "Pezzi ?";
 200 LET a$=INKEY$
 210 IF a$="" THEN GO TO 200
 220 IF INKEY$<>"" THEN GO TO 22
0
 225 BEEP .12,30
 230 LET b1=VAL a$: PRINT b1
 240 IF b1<=0 OR a(a1)-b1<0 THEN
 PRINT AT 0,0; PAPER 4;"errore "
: GO TO 190
 245 REM memorizza la mossa
 250 LET a(a1)=a(a1)-b1
 255 REM ordina le righe
 260 IF a(1)<a(2) THEN LET a1=a(
2): LET a(2)=a(1): LET a(1)=a1
 270 IF a(2)<a(3) THEN LET a1=a(
3): LET a(3)=a(2): LET a(2)=a1
 280 IF a(1)<a(2) THEN LET a1=a(
2): LET a(2)=a(1): LET a(1)=a1
 285 LET fl2=(fl2*(-1)+1)*fl1
 286 REM codifica la situazione
```

Seguito Listato 1.

```
 290 LET cod=100*a(3)+10*a(2)+a(
1)
 300 IF cod=1 THEN GO TO 1000: R
EM cancella la mossa perdente
 305 IF cod=0 THEN PRINT AT 0,0;
 PAPER 4;"mossa inaccettabile":
FOR i=1 TO 250: NEXT i: GO TO 10
0
 310 IF fl1=0 AND NOT cod=333 TH
EN GO SUB 2000: PRINT AT 20,1; P
APER 4;"Questa e' la tua mossa "
: BEEP .2,0: BEEP .4,-10: PAUSE
20
 320 GO SUB 3000: REM richiama
le mosse possibili
 350 IF m$(k)="      " THEN GO T
O 1000: REM cancella la mossa
 360 FOR i=1 TO 6: IF m$(k,i)<>"
" THEN NEXT i
 370 LET t=INT ((i-1)*RND)+1
 380 LET e$=m$(k,t): REM estrai
una mossa a caso
 385 IF fl2=0 THEN LET c$=e$
 386 IF fl2=1 THEN LET d$=e$
 390 IF e$="r" THEN GO TO 680
 400 GO SUB (500+(CODE e$-96)*10
)
 405 IF fl1=1 THEN GO TO 285
 410 GO SUB 2000: REM visualizza
situazione
 415 PRINT AT 20,1; PAPER 4; INK
 0;"Questa e' la mia mossa "
 416 PRINT AT 21,1; PAPER 4; INK
 0;"per favore inserisci la tua
"
 417 PAUSE 10
 420 GO TO 120
 500 REM mosse
 510 LET a(1)=3: LET a(2)=3: LET
a(3)=2: RETURN
 520 LET a(1)=3: LET a(2)=3: LET
a(3)=1: RETURN
 530 LET a(1)=3: LET a(2)=3: LET
a(3)=0: RETURN
 540 LET a(1)=3: LET a(2)=2: LET
a(3)=0: RETURN
 550 LET a(1)=3: LET a(2)=1: LET
a(3)=0: RETURN
 560 LET a(1)=3: LET a(2)=0: LET
a(3)=0: RETURN
 570 LET a(1)=3: LET a(2)=2: LET
a(3)=1: RETURN
 580 LET a(1)=3: LET a(2)=2: LET
a(3)=2: RETURN
 590 LET a(1)=3: LET a(2)=1: LET
a(3)=1: RETURN
 600 LET a(1)=2: LET a(2)=2: LET
a(3)=2: RETURN
 610 LET a(1)=2: LET a(2)=2: LET
a(3)=1: RETURN
 620 LET a(1)=2: LET a(2)=2: LET
a(3)=0: RETURN
 630 LET a(1)=2: LET a(2)=1: LET
a(3)=1: RETURN
 640 LET a(1)=2: LET a(2)=1: LET
a(3)=0: RETURN
 650 LET a(1)=2: LET a(2)=0: LET
a(3)=0: RETURN
 660 LET a(1)=1: LET a(2)=1: LET
a(3)=1: RETURN
 670 LET a(1)=1: LET a(2)=1: LET
a(3)=0: RETURN
 680 IF fl1=1 THEN LET fl2=fl2*(
-1)+1: GO TO 1000
 690 CLS : LET a(1)=1: LET a(2)=
0: LET a(3)=0: GO SUB 2000
 700 PRINT AT 10,10; PAPER 4; FL
```

Seguito Listato 1.

```
ASH 1;"HO VINTO": BEEP .5,20: BE
EP .5,25: BEEP .5,45: BEEP .5,10
: BEEP 1,15: GO TO 1060
1000 REM DELETE
1004 CLS
1005 IF fl1=0 THEN GO TO 1010
1006 IF fl2=0 THEN LET e$=c$
1007 IF fl2=1 THEN LET e$=d$
1008 IF e$="" THEN GO TO 1052
1010 FOR I=1 TO 18
1020 FOR K=1 TO 6
1030 IF m$(i,k)=e$ THEN LET m$(i
)=m$(i, TO k-1)+m$(i,k+1 TO )
1040 NEXT k
1050 NEXT i
1052 IF fl1=1 THEN RETURN
1055 PRINT AT 10,10; FLASH 1;" H
O PERSO "
1056 BEEP 1,-10: BEEP 2,-15.
1060 PRINT AT 20,1;"Vuoi giocare
 ancora (s or n)? ";
1062 LET a$=INKEY$: IF a$="" THE
N GO TO 1062
1065 IF INKEY$<>"" THEN GO TO 10
65
1070 IF a$<>"s" THEN STOP
1075 CLS
1080 GO TO 100
2000 REM VISUALIZZAZIONE
2010 PAPER 6: INK 2: CLS
2020 FOR k=1 TO 3
2030 FOR i=1 TO a(k)
2050 PRINT AT 20-6*k,3+7*i;"    "
2051 PRINT AT 21-6*k,3+7*i;"    "
2052 PRINT AT 22-6*k,3+7*i;"    "
2100 NEXT i
2105 IF a(k)>0 THEN PRINT AT 21-
6*k,1; PAPER 5; INK 0;k;"'riga"
2110 NEXT k
2115 INK 0: BEEP .2,3
2120 RETURN
3000 REM RICERCA
3005 RESTORE 3050
3010 FOR K=1 TO 18
3020 READ b
3030 IF cod=b THEN RETURN
3040 NEXT k
3050 DATA 333,233,133,33,23,13,3
,123,223,113,222,122,22,112,12,2
,111,11
4000 REM routine AUTO
4010 INPUT PAPER 4; INK 0;"Quant
e partite vuoi che giochi? ";num
4020 LET fl1=1: LET fl2=0
4030 FOR h=1 TO num
4035 CLS
4040 PRINT AT 0,0; PAPER 4;"Sto
giocando la ";h;"' partita": BEE
P .2,h AND h<70
4050 LET e$="": LET c$="": LET d
$="": FOR j=1 TO 3: LET a(j)=3:
NEXT j
4060 GO SUB 285: REM esegui ':IMB
LE
4070 NEXT h
4080 LET fl1=0
4090 RETURN
5000 REM definizione caratteri s
tampa pedine
5005 RESTORE 6000
5010 FOR i=1 TO 9: REM numero ca
ratteri
5020 READ c$
5030 FOR j=0 TO 7: REM numero by
te
5040 READ d
5050 POKE USR c$+j,d
5060 NEXT j: NEXT i
5070 RETURN
6000 DATA "a",0,1,7,15,31,63,63,
127
6010 DATA "b",126,255,255,255,25
5,255,255,255
6020 DATA "c",0,128,224,240,248,
252,252,254
6030 DATA "d",127,255,255,255,25
5,255,255,127
6040 DATA "f",254,255,255,255,25
5,255,255,254
6050 DATA "g",127,63,63,31,15,7,
1,0
6060 DATA "h",255,255,255,255,25
5,255,255,126
6070 DATA "i",254,252,252,248,24
0,224,128,0
6080 DATA "e",243,253,254,246,24
6,254,253,243
6090 REM fine programma
```

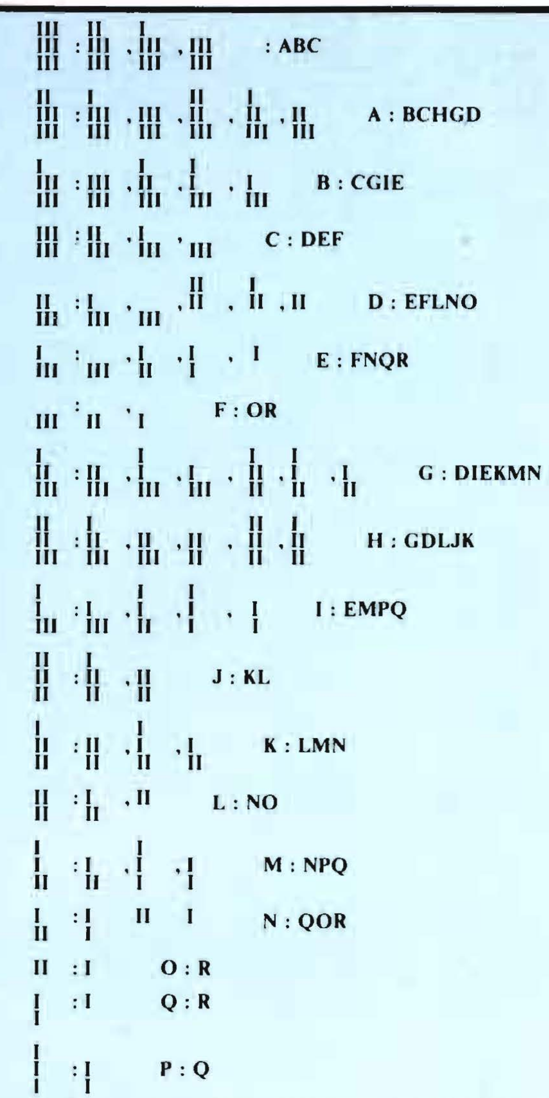


Figura 5. *Insieme iniziale delle mosse possibili.*

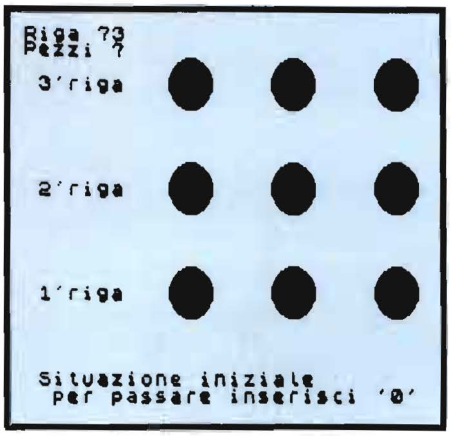


Figura 6. *Quadro iniziale del gioco del Nim.*

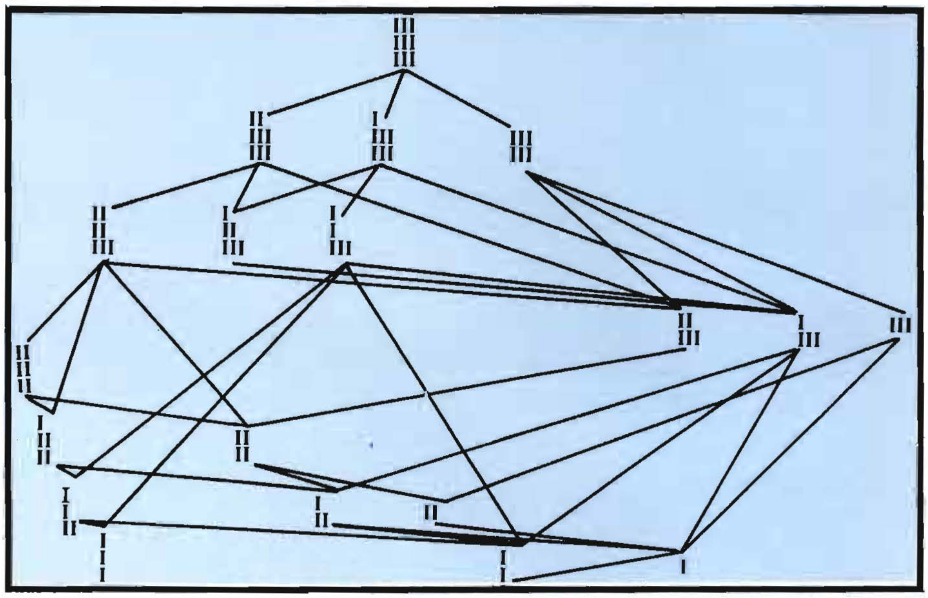


Figura 7. *Diagramma dello spazio degli stati.*

Type-Writer 1

— Parte seconda —

Un potente Text Editor per VIC 20

di *Alessandro Guida*

Questa è la seconda parte di un articolo, iniziato il mese scorso, che illustra un sofisticato programma in grado di trasformare il vostro VIC 20 in una moderna macchina per scrivere.
Vi sono i listati del programma e alcuni consigli per l'uso.
Per avere la descrizione completa dei comandi di questo text-editor dovrete, però, riferirvi al numero precedente di **Personal Software**.

Il programma

Poiché questo programma è stato realizzato per il VIC in versione da 16 Kbyte, è stato impossibile concentrare tutte le funzioni in un unico programma. Perciò Type-Writer 1 è stato diviso in due parti. La prima (listato 1), contiene tutti i DATA necessari a trasferire in memoria le routine in linguaggio macchina. Questa prima parte provvede, inoltre, a controllare se i DATA sono esatti, a proteggere la zona di memoria che contiene le routine in linguaggio macchina e a caricare la restante parte.
Dopo aver digitato l'intero listato 1 registratelo col nome Type-Writer 1. La seconda parte contiene, invece, il nucleo principale del programma, ed è in BASIC. Dopo aver digitato anche questa, registratela con il nome "TW1". Questo è molto importante, perché la prima parte del programma possa trovare e caricare

Listato 1. *Prima parte del programma Type-Writer 1.*
Il programma carica la routine in linguaggio macchina in memoria, e provvede a caricare anche la seconda parte del programma dal disco.

```
1 rem*************************
2 rem*                       *
3 rem*type-writer 1 x vic 20*
4 rem*   ♥                   *
5 rem*routine in l.m.        *
6 rem*                       *
7 rem*************************
10 restore
20 poke56,63:poke55,255:poke52,63:poke51,255:clr
30 n=22272:printchr$(14)
40 poke36879,8
50 print"█████    Attendere Prego!"
60 print"██Caricamento routine LM██e Programma Principale"
70 print"█████NON TOGLIERE IL DISCO █  DAL LETTORE FLOPPY  █"
80 fori=0to1139:reada:poken+i,a:tt=tt+a:next
90 iftt<>156580thenprint"█Errore nei data":end
100 clr:load"tw1",8
60000 data165,000,133,078,165,001,133,079,169,000
60010 data133,080,169,016,133,081,162,022,160,000
60020 data177,078,201,015,176,010,234,024,105,182
60030 data208,045,234,234,177,078,201,032,176,004
60040 data169,032,016,033,201,064,144,029,201,096
60050 data176,004,041,063,016,021,201,128,176,005
60060 data056,233,032,016,012,201,160,144,227,056
60070 data233,064,234,234,234,234,234,145,080,200
60080 data192,022,208,206,202,240,029,024,169,080
60090 data101,078,133,078,169,000,101,079,133,079
60100 data024,169,022,101,080,133,080,169,000,101
60110 data081,133,081,076,018,087,164,254,169,000
60120 data032,145,211,032,221,221,160,000,185,000
60130 data001,153,233,017,240,003,200,208,245,164
60140 data255,169,000,032,145,211,032,221,221,160
60150 data000,185,000,001,153,243,017,240,004,200
60160 data208,245,234,120,164,211,177,209,041,127
60170 data145,209,165,082,133,211,165,083,133,214
60180 data169,015,133,210,169,234,133,209,164,214
60190 data200,024,169,022,101,209,133,209,144,006
60200 data169,000,101,210,133,210,136,208,238,088
60210 data096,234,234,120,169,000,133,087,169,064
60220 data133,088,160,000,169,032,145,087,200,208
60230 data251,230,088,165,088,201,086,208,239,088
60240 data096,234,234,024,165,082,101,000,133,087
60250 data169,000,101,001,133,088,165,087,133,089
60260 data165,088,133,090,165,002,208,077,164,082
60270 data132,087,136,132,088,164,087,177,000,164
60280 data088,145,000,230,088,230,087,165,087,201
60290 data081,208,238,164,088,169,032,145,000,096
60300 data169,000,133,000,169,064,133,001,164,168
60310 data240,016,024,165,000,105,080,133,000,165
60320 data001,105,000,133,001,136,208,241,024,165
60330 data167,101,000,133,000,165,001,105,000,133
60340 data001,096,234,234,234,162,080,169,032,133
60350 data089,160,000,177,087,201,032,208,004,202
60360 data208,003,096,162,002,133,090,165,089,145
60370 data087,165,090,133,089,230,087,208,230,230
60380 data088,165,088,201,086,208,222,096,234,056
60390 data169,069,229,083,170,169,085,133,088,133
60400 data090,169,064,133,087,169,144,133,089,160
60410 data079,177,087,145,089,136,016,249,056,165
60420 data087,233,080,133,087,165,088,233,000,133
60430 data088,056,165,089,233,080,133,089,165,090
```

automaticamente la seconda.
Data la lunghezza dei due programmi è preferibile salvarli su disco. Naturalmente, nulla vieta di registrarli su nastro. In questo caso va corretta la linea 100 del listato 1, sostituendo un 1 all'8 che vi compare come ultimo carattere.
A questo punto, per usare il programma, sarà sufficiente dare:

LOAD "TYPE-WRITER 1", 8

ed il consueto RUN.

Il listato 1

Sul listato 1 c'è poco da dire, in quanto contiene solo un enorme elenco di DATA ed una routine che si occupa di leggerli e metterli in memoria a partire dall'indirizzo \$5700 (22272 decimale). Terminata l'operazione di memorizzazione viene caricata la seconda parte solo se i dati letti risultano esatti.
Tutti i DATA contengono i codici di numerose routine in linguaggio macchina che rendono il programma molto veloce.
Le routine più importanti sono quelle che gestiscono l'invio del testo alla stampante e la visualizzazione di una parte di esso sullo schermo.

Il listato 2

La seconda parte costituisce il programma vero e proprio che gestisce le varie opzioni a disposizione, e che si occupa della creazione e modifica del testo in memoria.
Il testo è memorizzato a partire dalla locazione \$4000 (16384 decimale), sotto forma di una pagina da 80 colonne per 70 righe. In pratica i vari caratteri del testo occupano lo stes-

Seguito listato 1.

```
60440 data233,000,133,090,202,208,218,160,079,169
60450 data032,145,089,136,016,251,096,234,234,024
60460 data165,087,105,080,133,089,165,088,105,000
60470 data133,090,166,083,160,079,177,089,145,087
60480 data136,016,249,024,165,087,105,080,133,087
60490 data165,088,105,000,133,088,024,165,089,105
60500 data080,133,089,165,090,105,000,133,090,232
60510 data224,069,208,216,160,079,169,032,145,087
60520 data136,016,251,096,234,234,234,234,166,253
60530 data164,254,169,000,133,253,169,064,133,254
60540 data152,240,016,024,169,080,101,253,133,253
60550 data169,000,101,254,133,254,136,208,240,138
60560 data024,101,253,133,253,169,000,101,254,133
60570 data254,164,255,177,253,217,119,002,208,004
60580 data136,208,246,096,230,253,208,239,230,254
60590 data165,254,201,086,208,231,169,255,133,255
60600 data096,234,234,234,234,234,234,120,165,254
60610 data141,060,003,165,255,141,061,003,024,165
60620 data252,101,254,133,254,169,000,101,255,133
60630 data255,160,000,162,048,177,254,201,032,240
60640 data020,202,208,009,234,024,165,252,105,002
60650 data133,255,096,198,254,208,234,198,255,208
60660 data230,160,001,177,254,153,255,001,201,032
60670 data240,007,169,032,145,254,200,208,240,024
60680 data173,060,003,105,080,133,254,173,061,003
60690 data105,000,133,255,024,165,251,101,254,133
60700 data254,169,000,101,255,133,255,160,000,185
60710 data000,002,201,032,240,005,145,254,200,208
60720 data244,024,152,101,251,133,255,088,096,234
60730 data234,234,234,234,234,234,169,064,133,083
60740 data169,000,133,082,160,000,177,082,201,015
60750 data048,048,164,252,177,082,201,032,208,040
60760 data136,196,251,240,035,177,082,201,032,240
60770 data245,132,255,162,002,164,251,177,082,201
60780 data032,240,035,200,196,255,048,245,202,208
60790 data240,240,007,132,254,234,234,234,234,234
60800 data024,169,080,101,082,133,082,169,000,101
60810 data083,133,083,201,086,208,183,096,132,254
60820 data165,252,133,000,133,001,230,001,164,000
60830 data177,082,164,001,145,082,198,001,198,000
60840 data165,000,197,254,016,238,230,255,165,255
60850 data197,252,240,202,164,254,200,200,208,173
60860 data234,234,234,169,001,166,254,160,001,032
60870 data186,255,165,255,162,000,160,002,032,189
60880 data255,169,000,032,213,255,096,234,234,234
60890 data169,001,166,254,160,001,032,186,255,165
60900 data255,162,000,160,002,032,189,255,169,000
60910 data133,251,169,064,133,252,169,251,162,255
60920 data160,085,032,216,255,096,169,004,162,004
60930 data160,007,032,186,255,169,000,133,000,032
60940 data189,255,032,192,255,162,004,032,201,255
60950 data169,000,133,254,169,064,133,255,160,080
60960 data132,002,160,000,177,254,201,015,048,060
60970 data076,020,091,234,177,254,032,210,255,200
60980 data196,002,208,246,192,080,240,005,169,015
60990 data032,210,255,230,000,165,000,197,169,240
61000 data015,024,169,080,101,254,133,254,169,000
61010 data101,255,133,255,208,198,164,001,169,010
61020 data032,210,255,136,208,250,032,231,255,096
61030 data201,014,240,004,169,032,208,194,169,041
61040 data133,002,208,186,032,228,255,016,181,048
61050 data231,169,000,133,167,133,168,032,044,088
61060 data165,167,133,255,165,168,133,254,032,000
61070 data087,032,228,255,240,251,201,017,208,010
61080 data165,168,201,048,240,241,230,168,016,223
61090 data201,029,208,010,165,167,201,080,240,029
61100 data230,167,016,209,201,145,208,008,165,168
61110 data240,215,198,168,016,197,201,157,240,001
61120 data096,165,167,240,202,198,167,016,184,169
61130 data000,133,167,240,178,234,234,234,255,255
```

Type-Writer 1

Listato 2. *Seconda parte del programma Type-Writer 1.*

```
2 poke45,046:poke47,046:poke49,046:poke46,57:poke48,57:poke50,57
5 l$="▀▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▀":dimtx$(5)
7 dimtb(3):tb(1)=10:tb(2)=30:tb(3)=50:nr=66
10 sp$="                    ":sr%=1
12 im=16384:is=4096:rm=69:lm=10
14 printchr$(14);chr$(8)
15 gosub1100:lf%=70-sr%
16 poke36879,8:print"▓▒"l$"Linea"tab(11)"Col.";
17 poke204,0:poke650,128
18 xc=lm:yc=0:xs=0:ys=0:goto80
20 ifpeek(653)and4then200
30 getr$:ifr$=""then20
31 ifr$=chr$(19)then18
32 ifr$=chr$(13)andyc<lf%thenxc=rm:goto72
33 ifr$=chr$(147)thensys(22483):sys(22272):goto18
34 ifr$=chr$(20)orr$=chr$(148)then600
35 ifr$="▐"then75
36 ifr$="▌"orr$="←"then1000
37 ifr$="◘"thenyc=yc-(yc<69):ys=ys-(ys<21):ym=yc-ys:goto120
38 ifr$="◙"thenyc=yc+(yc>0):ys=ys+(ys>0):ym=yc-ys:goto120
40 ch=asc(r$):ifch<32then20
45 ifch=157then80
50 ifch>127andch<161then20
60 ifch>191thench=ch-96
70 pokeim+xc+yc*80,ch
71 ifxc=rmandch<>32andyc<lf%then135
72 xc=xc+1:xs=xs-(xs<21):ifxc>rmandyc<lf%thenxc=lm:xs=2:yc=yc+sr%:ys=ys-sr%*(ys<
(22-sr%))
73 ifxc>rmthenxc=rm
74 goto80
75 xc=xc+1:xs=xs-(xs<21):ifxc=80andyc<69thenxc=0:xs=0:yc=yc+1:ys=ys-(ys<21)
77 ifxc=80thenxc=79
80 xm=xc-xs:ym=yc-ys
120 a=im+xm+ym*80:hb%=a/256:poke1,hb%:poke0,a-hb%*256
130 poke83,ys:poke82,xs:poke254,yc:poke255,xc:goto140
135 a=im+yc*80:hb%=a/256:poke255,hb%:poke254,a-hb%*256:sys(22878)
137 xc=peek(255):ifxc<rmthenyc=yc+sr%:xs=lm:ys=ys-sr%*(ys<(22-sr%)):goto80
138 xc=rm:goto72
140 sys22272
150 goto20
200 getr$:ch=peek(197)
210 ifch=18then500
220 ifch=53then12
230 ifch=0thenxc=tb(1)-1:xs=-21*(xc>57):goto75
240 ifch=56thenxc=tb(2)-1:xs=-21*(xc>57):goto75
250 ifch=1thenxc=tb(3)-1:xs=-21*(xc>57):goto75
260 ifch=12thenpoke83,yc:sys(22660):xc=lm:xs=0:goto80
270 ifch=34then2000
280 ifch=51then1300
290 ifch=43andxc=0thench=14:goto70
300 ifch=50then1400
310 ifch=42then1500
320 ifch=17thensys(23008):goto80
330 ifch=30thengosub1700:goto80
340 ifch=36thengosub3000:goto80
350 ifch=13thenpoke1,nf%:poke169,nr%:sys(23198):goto80
360 ifch=35andxc=0thench=12:goto70
370 ifch=27then450
380 ifch=41then2500
390 goto20
450 poke204,1:printl$"▓View ▓▒"
460 sys(23323)
470 xc=peek(167):yc=peek(168):xs=0:ys=0:poke204,0
480 printl$"Linea▒":getr$:goto80
500 fori=xcto79:pokeim+i+yc*80,32:next
510 sys(22272):goto20
600 poke2,abs(r$<>chr$(20)):a=im+yc*80:hb%=a/256:poke1,hb%:poke0,a-hb%*256
610 poke82,xc:sys(22513):ifr$=chr$(148)then80
1000 xc=xc-1:xs=xs+(xs>0):ifxc=-1andyc>0thenxc=79:xs=21:yc=yc-1:ys=ys+(ys>0)
1002 ifxc<0thenxc=0
1010 goto80
```

Seguito listato 2.

```
1100 print"██Definizione Parametri "
1170 fori=1to3:print"█TAB"i"        "tb(i);:input"█████";a$
1180 a=val(a$):ifa<0ora>79theni=11:goto1200
1190 tb(i)=a
1200 nexti:ifi>10then1100
1205 print"██Definizione Parametri "
1210 print"██Margini":print"███Sinistro:    "lm"█████";
1220 inputa$:a=val(a$):ifa<2ora>21then1205
1230 lm=a:poke251,lm:print"███Destro:    "rm"█████";
1240 inputa$:a=val(a$):ifa<58ora>78then1205
1250 rm=a:poke252,rm
1260 print"███Num. righe per foglio":print"     "nr"█████";
1270 inputn$:ifval(n$)<50orval(n$)>71then1260
1280 nr=val(n$):nr%=nr:nf%=67-6*(nr>66)-nr
1285 print"███Spaziatura lin.(1-2-3)":print"     "sr%"████";
1290 inputn$:ifval(n$)<1orval(n$)>3thenn$=str$(sr%)
1295 sr%=val(n$):poke22956,80*sr%:return
1300 poke83,yc:a=im+yc*80:hb%=a/256:poke88,hb%:poke87,a-hb%*256:sys(22730):xc=lm
:xs=0:goto80
1400 iftr=1then1410
1405 tr=1:yt=yc:poke4600,148
1407 fori=0to500:next:goto20
1410 tr=0:poke4600,32
1420 fori=0to79:pokeim+yc*80+i,peek(im+yt*80+i):next
1430 xc=lm:xs=0:goto80
1500 fori=0to21:poke4580+i,32:next
1510 printl$"FIND:"f$;
1520 getr$:ifr$=""then1520
1530 ifr$=chr$(13)then1570
1540 ifr$=chr$(20)andlen(f$)>0thenf$=left$(f$,len(f$)-1):print"█ █";:poke4586+le
n(f$),32:goto1520
1545 a=asc(r$):ifa<32ora>127anda<192then1520
1550 iflen(f$)=10then1520
1560 printr$;:f$=f$+r$:goto1520
1570 iff$=""thenf$=" "
1575 fori=1tolen(f$):a=asc(mid$(f$,i,1))
1580 ifa>192thena=a-96
1590 poke631+i,a:next:poke255,len(f$):poke253,xc:poke254,yc:sys(22800)
1600 ifpeek(255)=255then1650
1605 a=peek(254)*256+peek(253)-16384:yc=int(a/80):xc=a-yc*80
1607 xs=-xc*(xc<21)-10*(xc>20)-11*(xc>67):ys=-yc*(yc<21)-10*(yc>20)-11*(yc>45)
1610 ifyc>69thenyc=69
1620 fori=0to21:poke4580+i,32:next
1630 printl$"Linea"tab(11)"Col.";:goto75
1650 printl$"NOT FOUND!!! Again?";
1660 getr$:ifr$<>"y"andr$<>"n"then1660
1670 ifr$="n"then1620
1680 goto1500
1700 print"██    Elenco comandi    "
1710 print"█Comandi di tastiera:":print"──────────────────────"
1720 print"Home,Clr,CRSR:conserva     no la stessa fun-     zione originale."
1730 print"'←'= CRSR left
1740 print"DEL= Cancella il carat     tere sotto il cur     sore spostando a"
1750 print"     sinistra i carat-     teri seguenti del     la stessa riga."
1760 print"INST=Inserisce uno spa     zio spostando il      carattere sotto"
1770 print"     il cursore e i se     guenti a destra."
1780 print"██Premi un tasto";
1790 getr$:ifr$=""then1790
1800 print"█Comandi   diretti:":print"█(Premere con [CTRL] )"
1810 print"█/=istruzioni.":print"c=Passa al modo calco   latore (Premi █c█)."
1820 print"d=cancella un'intera    riga a partire dal    cursore."
1830 print"i=inserisce una riga    vuota sotto il cs."
1840 print"u=come d con la risali  ta delle righe suc-   cessive."
1850 print"b=inibisce l'allinea    mento a destra"
1860 print"@=definizione dei mar-  gini e dei TAB."
1870 print"██Premi un tasto o C";
1880 getr$:ifr$=""then1880
1890 ifr$<>"c"then1950
1900 print"██ Modo Calcolatrice  █"
1910 print"█+     =somma":print"-     =sottrazione":print"*,x  =moltiplicazione":p
rint"/,:  =divisione
```

Seguito listato 2.

```
1915 Print"CLR  =cancella ":Print"S    =inverte il segno":Print"E    =attiva l'
esPon."
1920 Print"%    =Percentuale":Print"=    =da il risultato":Print"A=torna al tex
t editor"
1930 Print"↑=come ? ma riPorta il  risultato a Partire   dal cursore."
1940 Print"Premi un tasto";
1945 getr$:ifr$=""then1945
1950 Print"Comandi   diretti:"
1955 Print"h=doPPia larghezza.":Print"1=Posiziona il cursore  al TAB 1"
1960 Print"2=come 1 ma va al TAB2";:Print"3=va al TAB 3."
1970 Print"t=trasferisce una riga  da un Punto ad un al- tro del testo."
1980 Print"f=cerca un vocabolo      nel testo."
1982 Print"m=gestione memoria di    massa (Premi m)."
1984 Print"a=allineamento del tes  to col margine dxt"
1966 Print"P=stamPa "
1988 Print"s=stamPa continua":Print"v=VIEW/sPosta finestra"
1991 Print"Premi un tasto o M";
1992 getr$:ifr$=""then1992
1994 ifr$="m"thenzz%=1:goto3000
1995 return
2000 fori=0to3:Poke4580+i,160:next:fori=4to17:Poke4580+i,32:next:fori=18to21:Pok
e4580+i,160:next
2010 n$(0)="":n$(1)="":tt$="":vs$="0":Poke204,1:c=0:ex=0:dc=0
2020 ifval(tt$)>9e19thengoto2400
2025 Printl$tab(3)right$(sP$+vs$,15);
2030 getr$
2040 ifr$<"0"orr$>"9"then2060
2045 ifc=0andn$(0)=tt$thenn$(0)=""
2050 iflen(n$(c))<15andex<2thenn$(c)=n$(c)+r$:vs$=n$(c):ex=ex-(ex<>0):goto2020
2060 ifr$=chr$(19)orr$=chr$(147)thenn$(c)="":vs$="0":ex=0:dc=0:goto2020
2065 ifr$=chr$(222)thenn$(c)=" 3.14159":vs$=n$(c):goto2020
2070 ifr$<>"+"andr$<>"-"andr$<>"*"andr$<>"x"andr$<>"/"andr$<>":"andr$<>"%"then20
85
2080 ifc=0thenc=1:oP$=r$:vs$="0":ex=0:dc=0:Poke4599,asc(r$)+128:goto2020
2085 ifr$<>"="then2200
2087 Poke4599,160
2090 n0=val(n$(0)):n1=val(n$(1)):ifoP$="+"thentt$=str$(n0+n1):ex=0:dc=0:goto2190
2100 ifoP$="-"thentt$=str$(n0-n1):goto2190
2110 ifoP$="*"oroP$="x"thentt$=str$(n0*n1):goto2190
2120 ifoP$="/"oroP$=":"thentt$=str$(n0/n1):goto2190
2130 ifoP$="%"thentt$=str$(n0/100*n1):goto2190
2190 n$(0)=tt$:n$(1)="":c=0:vs$=tt$:goto2020
2200 ifr$="e"andex=0thenex=1:n$(c)=n$(c)+"e+":vs$=n$(c):goto2020
2210 ifr$<>"s"then2300
2220 ifex>0then2250
2230 ifleft$(n$(c),1)<>"-"thenn$(c)="-"+n$(c):vs$=n$(c):goto2020
2240 n$(c)=right$(n$(c),len(n$(c))-1):vs$=n$(c):goto2020
2250 fori=1tolen(n$(c))+1
2260 ifmid$(n$(c),i,1)<>"e"then2290
2270 ifmid$(n$(c),i+1,1)="+"thenn$(c)=left$(n$(c),i)+"-"+mid$(n$(c),i+2):vs$=n$(
c):i=20:goto2290
2280 n$(c)=left$(n$(c),i)+"+"+mid$(n$(c),i+2):vs$=n$(c):i=20
2290 next:goto2020
2300 ifr$="."andex=0anddc=0thendc=1:n$(c)=n$(c)+".":vs$=n$(c):goto2020
2310 ifr$="?"thenPoke204,0:goto2350
2320 ifr$<>"↑"then2030
2330 Poke204,0:iftt$=""thentt$=" 0"
2340 fori=1tolen(tt$):Pokeim+xc+yc*80+i-1,asc(mid$(tt$,i,1)):next
2350 Print""l$"Linea"tab(11)"Col.";
2360 goto130
2400 Printl$tab(3)"OVERFLOW  ERROR";
2410 getr$:ifr$=""then2410
2420 goto2000
2500 Print"   StamPa continua    "
2510 Print""l$"*-Per terminare "
2520 fori=0to5:Print"Testo n."chr$(49+i)"":Print""left$(tx$(i)+sP$,18)
2530 inPut"";tx$(i):iftx$(i)="*"theni=5
2535 iflen(tx$(i))>10then2530
2537 next
2540 Print"StamPa continua testi "
2550 fori=0to5:Print""chr$(49+i)") "tx$(i):next
```

Seguito listato 2.

```
2560 print"▒"l$"▒va bene?▒";
2570 getr$:ifr$<>"s"andr$<>"n"then2570
2580 ifr$="n"then2500
2590 print"▒"l$"▒Numero copie(1-9)?▒";
2600 getr$:ifr$<"1"orr$>"9"then2600
2610 print"▒"l$"Premere F1 per finire▒"
2620 forqc=1toval(r$)
2630 forlo=0to5:iftx$(lo)="*"ortx$(lo)=""thenlo=6:goto2750
2640 print"▒▒  Caricamento Testo    ▒":print"▒"tx$(lo)
2650 poke255,len(tx$(lo))
2670 fori=1tolen(tx$(lo)):poke511+i,asc(mid$(tx$(lo),i,1)):next
2680 poke254,8:sys(23135):gosub5000
2690 ifer>20then2750
2700 ifpeek(197)=39thenlo=6:qc=9:goto2750
2710 print"▒▒  Stampa   Testo       ▒"
2720 poke1,nf%:poke169,nr%:sys(23198)
2745 ifpeek(197)=39thenlo=6:qc=9
2750 nextlo,qc
2760 goto16
3000 print"▒▒Gestione Pagina Testo"
3010 print"▒Comandi:":print"─────────"
3020 print"L-Carica in memoria"
3030 print"S-Registra"
3040 print"▒Comandi disco:":print"───────────────"
3050 print"R-Cambia nome ad un te  sto gia' sul disco"
3060 print"F-Formatta dischetto"
3070 print"D-Cancella un testo     dal disco"
3075 print"$-Mostra il contenuto   del dischetto"
3080 print"Q-torna al text editor"
3090 print"▒▒premi un tasto▒";
3100 getr$:ifr$=""then3100
3105 :ifzz%=1thenzz%=0:goto16
3110 ifr$="q"then16
3120 ifr$="f"then3500
3130 ifr$="d"then3600
3140 ifr$="r"then3700
3150 ifr$="l"then3800
3160 ifr$="s"then4000
3170 ifr$="$"then4500
3180 goto3100
3500 print"▒▒  Formattazione disco "
3510 print"▒Nome (max 16 car.):":inputnm$:iflen(nm$)>16then3500
3515 ifnm$=""then3000
3520 print"▒ID (2 caratt.):":inputid$:iflen(id$)<>2then3500
3530 print"▒▒  Inserire dischetto  ▒      da formattare"
3540 print"▒  e premere [RETURN]"
3550 getr$:ifr$<>chr$(13)then3550
3560 open15,8,15,"n0:"+nm$+","+id$
3570 print"▒▒▒▒▒▒▒▒Attendere prego!":close15
3580 gosub5000:goto3000
3600 print"▒▒ Cancellazione Testo  "
3610 print"▒▒Nome (max 10 car.):"
3620 inputnm$:iflen(nm$)>10then3600
3630 ifnm$=""then3000
3640 open15,8,15,"s0:"+nm$:close15
3650 gosub5000:goto3000
3700 print"▒▒    Reintestazione     "
3710 print"▒▒Vecchio nome (max 10)":inputvm$:iflen(vm$)>10then3700
3720 ifvm$=""then3000
3730 print"▒Nuovo nome (max 10)":inputnm$:iflen(nm$)>10then3000
3740 ifnm$=""then80
3750 open15,8,15,"r0:"+nm$+"="+vm$:close15
3760 gosub5000:goto3000
3800 print"▒▒**Caricamento  Testo**"
3810 print"▒▒Nome (max 10 car.)"
3820 inputnm$:iflen(nm$)>10then3800
3830 ifnm$=""then3000
3840 print"▒▒▒▒Registr. o ▒F▒loppy ?"
3850 getr$:ifr$<>"r"andr$<>"f"then3850
3860 poke255,len(nm$)
3870 fori=1tolen(nm$):poke511+i,asc(mid$(nm$,i,1)):next
```

PERSONAL SOFTWARE

Type-Writer 1

```
3880 poke254,-(r$="r")-8*(r$="f")
3890 sys(23135):gosub5000:goto16
4000 print"*Registrazione  Testo*"
4010 print"Nome (max 10 car.)"
4020 inputnm$:iflen(nm$)>10then4000
4030 ifnm$=""then3000
4040 print"Nome gia' esistente?"
4050 getr$:ifr$<>"s"andr$<>"n"then4050
4060 ifr$="s"thennm$="@0:"+nm$
4070 print"Registr. o Floppy ?"
4080 getr$:ifr$<>"r"andr$<>"f"then4080
4090 poke255,len(nm$)
4100 fori=1tolen(nm$):poke511+i,asc(mid$(nm$,i,1)):next
4110 poke254,-(r$="r")-8*(r$="f")
4120 sys(23162):gosub5000:goto3000
4500 print" Contenuto  dischetto ":n=0:open14,8,0,"$0"
4510 fori=0to6:gosub4700:next:dn$="":fori=0to18:gosub4700:dn$=dn$+a$:next
4550 print"Vol:"dn$:print"contenente:":print"————————————"
4560 fori=1to20:gosub4700:ifa$=chr$(0)theni=100
4580 next
4590 fori=1to4:gosub4700:if(st)thenclose14:goto4800
4600 next:f$="":j=0:fori=1to30:gosub4700:ifa$<>chr$(34)then4670
4650 ifj=1theni=100:goto4680
4660 j=1:goto4680
4670 ifj=1thenf$=f$+a$
4680 next:n=n+1:iff$<>""thenprintn"";
4690 printf$:goto4560
4700 get#14,a$:ifa$=""thena$=chr$(0)
4710 return
4800 print"Premi un tasto";
4810 getr$:ifr$=""then4810
4820 goto3000
5000 open15,8,15
5010 input#15,er,er$,et,es:close15
5020 ifer<20thenreturn
5025 printl$"ERRORE UNITA' DISCO  ";
5030 printer;er$;:printl$et,es;
5040 getr$:ifr$=""then5040
5050 return
```

so posto che occuperebbero su di un grande schermo dalle dimensioni viste prima. Naturalmente questo non è il sistema più efficiente per conservare dei testi in memoria.

Purtroppo, però, abbiamo dovuto optare per questo tipo di organizzazione a causa delle ridotte dimensioni dello schermo del VIC. Infatti, disponendo di sole 23 righe per 22 colonne, era assolutamente necessario realizzare un sistema di visualizzazione con scrolling estremamente efficiente e veloce, tenendo presente anche che buona parte del programma, essendo scritta in BASIC, sarebbe stata un po' lenta. Il tipo di memorizzazione visto prima, invece, rende molto semplice la visualizzazione di una parte della pagina, detta finestra.

Per gestire questa finestra, sono utilizzate tre coppie di variabili:

XC, XY Indicano le coordinate del cursore sul foglio;

XS, YS Indicano le coordinate del cursore sullo schermo;

XM, YM Indicano le coordinate dell'angolo in alto a sinistra della finestra sul foglio.

Buona parte del programma è dedicata alla gestione di queste 6 variabili.

Nel listato 2 si distinguono anche, dalla linea 1700 alla 2000, le varie videate di aiuto che forniscono il riepilogo dei comandi, attivate con la pressione dei tasti CTRL '/'.

Dalla linea 30 alla 40, vengono interpretati i comandi di tastiera (CLR, INST, DEL, ecc.).

Dalla linea 200 alla 390, invece, vengono analizzati i tasti premuti insieme al CTRL per eseguire i vari comandi a questi associati (vedi riquadro 1).

È possibile aggiungere delle nuove linee tra la 200 e la 390, per intercettare altri tasti ed aggiungere, così, nuove funzioni al nostro text-editor. In questo caso, però, è necessario modificare anche la linea 2. Questa serve ad aggiornare i vettori in pagina zero, contenenti la fine del programma BASIC e l'inizio delle variabili, poiché questo non viene fatto dal computer dopo l'istruzione di LOAD del listato 1.

Per conoscere i valori da introdurre in tale linea basta battere in modo diretto:

PRINT PEEK (45), PEEK (46)

subito dopo avere aggiunto le nuove linee al programma e prima di dare il RUN.

Si tenga anche presente che di memoria libera ne resta molto poca, per cui le modifiche apportate al programma non lo devono allungare molto.

Type-Writer 1

Suggerimenti sull'uso del programma

Diamo qui alcune precisazioni in aggiunta a quanto già detto nella prima puntata sui vari comandi di Type-Writer 1.

1) In modo calcolatore si hanno a disposizione anche i seguenti comandi: S, E, π, CLR; che vanno dati senza l'uso del tasto CTRL.

"E" attiva la notazione esponenziale (per esempio 1000 = 1E + 3).

"S" cambia il segno del numero o dell'esponente. "π" che si ottiene premendo il tasto SHIFT + "↑", fa apparire sullo schermo il valore 3,14159. "CLR", infine, cancella l'ultimo numero impostato.

2) In fase di definizione dei parametri iniziali è possibile scegliere anche la spaziatura tra le linee, che può essere 1, 2 o 3.

3) Come abbiamo già detto il text-editor gestisce una pagina di testo da 80 colonne per 70 righe. Il formato normale di stampa è, invece, di 80 x 66. Questo vuol dire che le ultime 4 righe generalmente non vengono stampate. Le 4 righe superflue sono comunque molto utili per evitare che utilizzando per errore la funzione di inserimento linea le ultime righe del testo vadano perse.

4) Il tasto F1 serve a terminare in qualsiasi momento una stampa.

5) La funzione VIEW permette uno scrolling molto veloce dello schermo, ed è utile per rileggere il testo prima di stamparlo. In particolare lo scrolling verso destra è continuo, per facilitare al massimo la lettura del testo.

6) Nell'impostare i parametri iniziali bisogna indicare il numero di righe che si vogliono stampare in un foglio.

Se tale numero è minore o uguale a 66 viene supposta la presenza di fogli da 66 linee altrimenti da 72.

7) La funzione FIND, per rintracciare dei caratteri all'interno della pagina, esegue la ricerca dal cursore in poi.

8) La funzione SEQ di stampa continuata, permette di stampare una sequenza definita di testi caricati dal disco. L'operazione può essere interrotta con il tasto F1.

La funzione SEQ non funziona con il registratore.

9) Se in qualsiasi momento non si ricordano più i comandi a disposizione si può premere CTRL?. Verrà visualizzato l'elenco dei comandi *senza perdere* il testo su cui si stava lavorando.

Le figure 1, 2 e 3 illustrano un esempio di composizione di una lettera. In essa viene fatto uso anche della funzione Calcolatore, e dell'allineamento del testo a destra. ■

RIASSUNTO DEI COMANDI PIU' IMPORTANTI

Comandi diretti (premere CTRL)

/ (?)	IST	H	HIGH	U	UP
A	AUTO	I	INS	V	VIEW
B	BLOC	M	MEM	@	PAR
C	CALC	P	PRINT	1	TAB1
D	DEL	S	SEQ	2	TAB2
F	FIND	T	TRANS	3	TAB3

Comandi indiretti (senza il CTRL)

Per la memoria di massa

$	DIR	L	LOAD	S	SAVE
D	DEL	Q	QUIT	↑	ENTER
F	FORM	R	REN		

In Modo Calcolatore

X, ★	MOLT	+	ADD	−	SUB
: , /	DIV	%	PERC	=	RES
Q	QUIT	!	ENTER	E	EXP
S	SIGN	π	PI	CLR	CLR

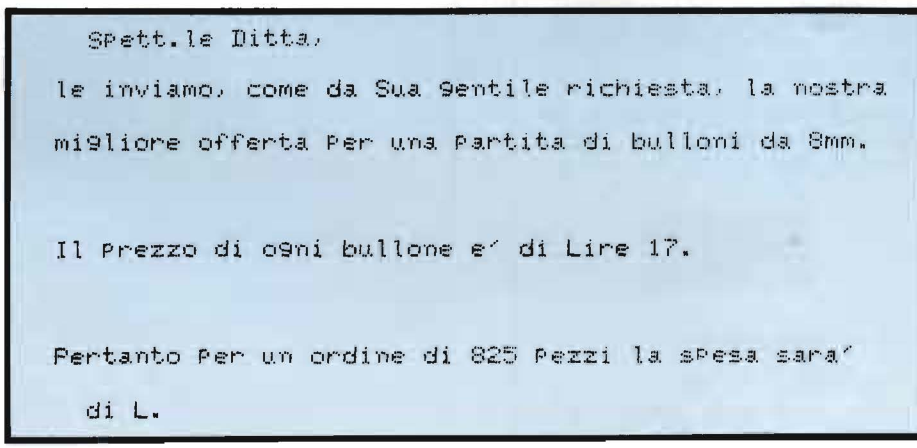


Figura 1. *Esempio di testo composto con il nostro programma di Text Editor. Per calcolare l'importo totale indicato in questo esempio ci serviremo della funzione Calcolatore. Per passare a tale funzione basta premere i tasti CTRLC.*

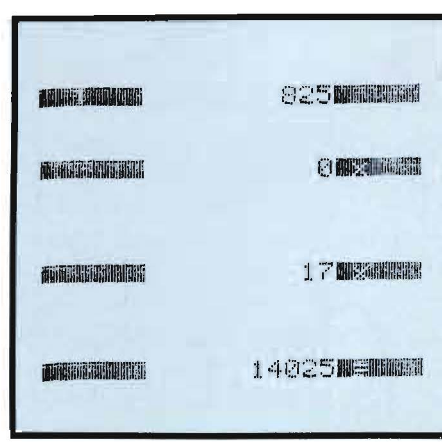


Figura 2. *Esempio di calcolo utilizzante la funzione Calcolatore di Type-Writer 1.*

```
  Spett.le Ditta,
le inviamo, come da Sua gentile richiesta, la nostra
migliore offerta per una partita di bulloni da 8mm.

Il prezzo di ogni bullone e' di Lire 17.

Pertanto per un ordine di 825 pezzi la spesa sara'
  di L. 14025 + IVA.

In attesa di un suo gradito ordine le porgiamo i nostri
migliori saluti.
```

Figura 3. *Durante i calcoli l'ultima linea dello schermo apparirà come in figura 2. Al termine delle operazioni è sufficiente premere il tasto "↑" perchè il risultato sia trascritto dove avevamo lasciato il cursore, come si vede in questa figura. Si può quindi terminare la stesura della lettera.*

```
  Spett.le Ditta,
le   inviamo,   come   da   Sua   gentile   richiesta,
la   nostra   migliore   offerta   per   una   partita
di   bulloni   da   8mm.

Il prezzo di ogni bullone e' di Lire 17.

Pertanto   per   un   ordine   di   825   pezzi   la   spesa
sara'   di L. 14025 + IVA.

In   attesa   di   un   suo   gradito ordine le porgiamo
i nostri   migliori saluti.
```

Figura 4. *Dopo aver terminato la scrittura, possiamo eseguire l'allineamento a destra del testo premendo contemporaneamente i tasti CTRL e A.*

Death per C 64

Un divertente gioco direttamente dalla... Transilvania

di *Luca Marras*

Death è un videogame redatto interamente in BASIC. Nonostante questo la velocità di esecuzione è elevata. Come vedremo più avanti, infatti, ho utilizzato al massimo le capacità grafiche e sonore del C 64, stando però attento a non sacrificare la velocità, componente fondamentale di ogni buon gioco.
Il tema di questo videogame è piuttosto originale, avendo come precedente solo il Dracula della Imagic per la console Intellivision. In ogni caso, non è mai stato sfruttato su di un personal. Si tratta, infatti, di aiutare un vampiro a superare delle vere e proprie pioggie di nemici, rappresentate di volta in volta da teschi coloratissimi o tombe in technicolor! Il nostro povero vampiro, però, avrà anche un amico... un pipistrello che lo aiuterà a rifocillarsi. Ma c'è un inconveniente: questo pipistrello è dispettoso e si diverte a sparire dallo schermo proprio quando c'è più bisogno di lui... Sopravvivere sino alla fine di ogni livello non è facile come sembra: ve ne accorgerete giocando!
Per superare il primo livello, dovrete evitare lo scontro con i 50 teschi che salgono instancabilmente verso l'alto. Fatto ciò, sarete festeggiati da un'allegra musichetta, mentre due pipistrelli vi porteranno un bonus di ben 1000 punti. Al secondo livello dovrete combattere contro 60 tombe... al terzo, i vostri nemici saranno 70 e così via. Alla fine di ogni livello avrete il consueto bunus. Esso è molto utile, poichè ogni 5000 punti vi sarà regalata una vita extra (la dotazione di partenza è di tre vite). Ogni volta che ne perdete una, saranno visualizzati il punteggio, il numero di vite rimaste e il numero di nemici che dovrete ancora evitare.
Al termine del gioco, se avrete realizzato un record il computer vi chiederà il vostro nome, che apparirà assieme al vostro punteggio nello schermo di presentazione del gioco.
Il movimento del vampiro è unicamente orizzontale, e viene comandato dalla tastiera attraverso i seguenti pulsanti:

L = Sinistra
; = Destra

Inoltre, viene utilizzato un altro tasto ("A") per catturare il pipistrello. Esso va premuto ripetutamente quando vi trovate sopra il pipistrello, e vi procurerà 20 punti ad ogni tocco.
Vediamo ora come è organizzato il programma
La linea 9 disinserisce il tasto RUN STOP. Questo è solo un accorgimento pratico per evitare un indesiderato BREAK del programma. Se volete reinserire il RUN STOP basterà cancellare questa linea.
Le linee da 10 a 70 preparano l'esecuzione del gioco e creano la grafica e gli sprite. Esse vengono eseguite una sola volta (subito dopo il RUN). Hanno anche il compito di leggere i DATA.
Le linee da 80 a 97 preparano lo schermo di presentazione e mandano alla subroutine 13000 - 13070, che crea la scritta Death e stampa il massimo punteggio.
Le linee da 100 a 190 costituiscono il cuore del programma: controllano se si è premuto un tasto direzionale (nel qual caso mandano alla subroutine 2000 - 2100), se c'è stata una collisione, e muovono i caratteri grafici verso l'alto.
Le linee da 1000 a 1110 creano il disegno dei teschi.
Le linee da 1500 a 1730 creano i tre sprite del gioco (vampiro, pipistrello e croci).
Le linee da 1800 a 1897 creano la figura delle tombe che compaiono nei livelli pari.
Le linee 1900 e 1910 creano la musichetta che accompagna il bonus alla fine di ogni livello.
Le linee da 2000 a 2090 costituiscono la subroutine che, dopo aver controllato quale tasto è stato premuto, muove il vampiro in senso orizzontale. Questa subroutine gestisce anche la funzione del tasto "A". Inoltre, attraverso la variabile casuale RN, stabilisce anche se il pipistrello deve apparire sullo schermo (subroutine 3000 - 3100) o sparire (3200 - 3220).
La linea 2500 è una piccola subroutine che genera un suono acuto se si preme un qualsiasi tasto. Questo suono è più lungo se il pulsante premuto è un tasto direzionale ("L" o ";").
Le linee da 3000 a 3100 stabiliscono in quale posizione deve apparire il pipistrello sullo schermo.
Le linee 4000 e 4100 fanno aumentare il punteggio di 20 punti e generano un suono acuto (vedi meccanismo di gioco - funzione del tasto "A").
La subroutine 7000 - 7070 crea lo schermo in cui sono visualizzati il punteggio della partita, il punteggio massimo e il numero del livello da giocare. Questo schermo appare prima dell'inizio di un nuovo livello.
Le linee da 8000 a 8500 determinano la perdita di una vita e, qualora questa fosse l'ultima rimasta, mandano

Death per C 64

Listato 1. *Il programma BASIC Death.*

```
0 REM ********************************
    *********              *********
1 REM *********    "DEATH"    *********
    *********              *********
2 REM *********      BY       *********
    *********              *********
3 REM *********  LUCA MARRAS  *********
    *********              *********
4 REM ********************************
9 POKE808,225:REM DISINSERIMENTO DEL
    TASTO RUN STOP
10 POKE53280,0:POKE53281,0:PRINT"[<1CLR>
]":V=53248:X=150:Y=140:DIMDI$(12,2):POKE
V+23,4
13 GA$="GAME OVER":PRINT"[<9CRSR D>][<1G
RN>]VIDEOGAME FOR THE COMMODORE 64"
15 POKEV+41,5:X2=X:POKEV+29,12:POKE2043,
13:POKEV+7,140:POKEV+42,11:POKEV+27,255
17 DIMDI(62):DIMD2(62):DIMD3(62):POKEV+2
1,0
20 FORP=1TO12:READDI$(P,0):NEXT:POKEV+43
,4
30 FORP=0TO62:READDI(P):POKE832+P,DI(P):
NEXT
35 FORP=0TO62:READD2(P):POKE896+P,D2(P):
NEXT
37 FORP=0TO62:READD3(P):POKE960+P,D3(P):
NEXT
38 FORP=1TO12:READDI$(P,2):NEXT:FORP=1TO
9:READN1(P),N2(P),DU(P):NEXT
40 POKE2042,13:POKE2043,15:POKE2044,15
50 SI=54272:FORP=0TO2:FL(P+1)=SI+P*7:FH(
P+1)=SI+P*7+1:TL(P+1)=SI+P*7+2
55 TH(P+1)=SI+P*7+3:W(P+1)=SI+P*7+4:AA(P
+1)=SI+P*7+5:HH(P+1)=SI+P*7+6:NEXT
57 FORP=1TO3:POKEW(P),0:NEXT
60 VL=SI+24:POKEVL,15:POKEAA(1),0*16+6:P
OKEHH(1),0*16+0:POKETL(2),15
65 POKEFH(1),4:POKEFL(1),90:POKETH(2),15
:POKEAA(2),0*16+8:POKEHH(2),0*16+0
67 POKEFH(2),69:POKEFL(2),157:POKEAA(3),
0*16+10:POKEHH(3),4*16+2
70 POKETL(3),100:POKETH(3),10:POKEFL(3),
100:POKEFH(3),20
80 Q=0:NT=12:NS=0:I=2:SG=1:UD=-1:RE=5000
:AU=0
81 PRINT"[<1CLR>][<21CRSR D>][<1ARA>](C)
 1984 L.M.S. - DESIGNED BY L. MARRAS"
82 PRINT"[<1HOME>][<18CRSR D>][<1CYN>]PU
SH 'F1' TO BEGIN":POKE198,0
83 GOSUB14000:GOSUB13000:GOTO15000
84 GOTO83
85 GOSUB7000
90 PRINT"[<1CLR>]":POKEW(3),0:POKEW(3),1
29:FORP=1TO10:IFPEEK(V+31)=4THENK=K
95 NEXT:PRINT"[<20CRSR D>]":POKEV+16,0:P
OKE198,0:X=150:X2=X:POKEV+4,X2
97 POKEV+21,4
100 CO=INT(RND(0)*15+1)
110 ONCOGOSUB30000,30010,30020,30030,300
40,30050,30060,30070,30080,30090,30100
115 IFCO<12THEN130
120 ONCO-11GOSUB30110,30120,30130,30140
130 SP=INT(RND(1)*31+1)
140 FORP=1TONT
150 PRINTTAB(SP);
155 GETA$:IFA$<>""THENGOSUB2000
170 POKEV+4,X2:POKEV+5,Y:POKEV+41,INT(RN
D(2)*15+1)
175 IFPEEK(V+31)=4THEN8000
177 POKEW(1),0:POKEW(1),33
180 PRINTDI$(P,1+UD):NEXT
185 NS=NS+1:IFNS=50+AUTHEN11000
190 GOTO100
1000 DATA"[<1RVS>][<2CRSR R>]     "
1010 DATA"[<1RVS>][<1CRSR R>]       "
1020 DATA"[<1RVS>]!        [<1RVS OFF>]!
"
1030 DATA"[<1RVS>][<1CRSR R>] [<1CRSR R
>]  [<1CRSR R>] "
1040 DATA"[<1RVS>][<1CRSR R>]  [<1RVS
OFF>][<1CHR$(169)>][<1CHR$(223)>][<1RVS
>]  "
1050 DATA"[<1RVS>][<2CRSR R>]     "
1060 DATA"[<1RVS>][<2CRSR R>]![<2CRSR R
>][<1RVS OFF>]!"
1070 DATA"  [<1CHR$(182)>][<1RVS>]  [<1
RVS OFF>][<1CHR$(181)>]"
1080 DATA" "
1090 DATA" "
1100 DATA" "
1110 DATA" "
1500 DATA0,0,0,0,255,0,63,255,252,127,25
5,254,125,255,190
1510 DATA126,255,126,127,126,254,124,189
,62,124,219,62,31,255,252
1520 DATA15,255,240,7,255,224,3,66,192,3
,66,192
1530 DATA3,66,192,3,66,192,1,0,128,1,255
,128,0,126,0,0,0,0,0,0,0
1600 DATA0,60,0,0,60,0,0,60,0,0,60,0,0,6
0,0
1610 DATA31,255,248,63,255,252,31,255,24
8,0,60,0,0,60,0
1620 DATA0,60,0,0,60,0,0,60,0,0,60,0,8,6
0,16
1630 DATA8,60,16,28,60,56,8,60,16,8,60,1
6,8,60,16,8,60,16
1700 DATA0,0,0,0,0,0,0,0,0,0,0,0,224,24,
7
1710 DATA248,60,30,252,60,62,62,60,124,6
```

Seguito listato Death.

```
3,24,252,31,255,248
1720 DATA31,255,248,31,255,248,7,255,224
,7,255,224,3,231,192
1730 DATA1,195,128,1,129,128,0,0,0,0,0,0
,0,0,0,0,0,0
1800 DATA"[<1RVS OFF>][<3CRSR R>][<1CHR
$(219)>]"
1810 DATA"[<1RVS OFF>][<3CRSR R>][<1CHR
$(194)>]"
1820 DATA"[<1RVS OFF>][<2CRSR R>][<1CHR
$(213)>][<2CHR$(195)>][<1CHR$(201)>]"
1830 DATA"[<1RVS OFF>][<2CRSR R>][<1CHR
$(194)>]**[<1CHR$(194)>]"
1840 DATA"[<1RVS OFF>][<2CRSR R>][<1CHR
$(194)>]  [<1CHR$(194)>]"
1850 DATA"[<1RVS OFF>][<2CRSR R>][<1CHR
$(194)>]  [<1CHR$(194)>]"
1860 DATA"[<1RVS>][<1CRSR R>][<1CHR$(16
9)>]    [<1CHR$(223)>]"
1870 DATA"[<1RVS>][<1CHR$(169)>]        [
<1CHR$(223)>]"
1880 DATA"[<1RVS OFF>][<1CHR$(204)>][<6
CHR$(175)>][<1CHR$(186)>]"
1890 DATA" "
1895 DATA" "
1897 DATA" "
1900 DATA21,237,400,17,103,400,21,237,40
0,26,20,400
1910 DATA29,69,100,26,20,100,23,59,100,1
9,137,100,17,103,600
2000 IFA$="L"THENK=-10:GOSUB2500
2010 IFA$=";"THENK=+10:GOSUB2500
2020 X=X+K
2030 IFX<30THENX=X-K
2035 X2=X
2040 IFX>255THENX2=X-255:POKEV+16,21:EN=
1:IFX2>35THENX=X-K:X2=X-255
2050 IFEN=1ANDX<=255THENX2=X:POKEV+4,X2:
POKEV+16,0:EN=0
2060 RN=INT(RND(2)*50+1):IFRN=26THENGOSU
B3000
2070 IFRN=25THENGOSUB3200
2075 IFA$<>"A"THEN2090
2080 IFPEEK(V+30)>0THENGOSUB4000:GOTO209
0
2085 POKEW(3),129
2090 K=0:GOSUB2500
2100 RETURN
2500 POKEW(2),0:POKEW(2),65:RETURN
3000 POKEV+6,INT(RND(2)*220+30):POKEV+21
,12
3100 RETURN
3200 POKEV+21,4:POKEW(3),129:FORU=1TO10
3210 IFPEEK(V+31)=4THENK=K
3220 NEXT:RETURN
4000 Q=Q+20:POKEW(3),0:POKEW(3),33
4100 RETURN
7000 POKEV+21,0:OC=1:POKEW(2),0:POKEW(2)
,33
7010 PRINT"[<1CLR>][<1YEL>][<22CRSR R>][
<1RVS>]STAGE"SG:PRINT
7020 PRINT"[<22CRSR R>][<1ARA>]READY !"
7030 PRINT"[<1HOME>][<1BLU>][<1CHR$(207)
>][<19CHR$(183)>][<1CHR$(208)>]"
7035 PRINT"[<1CHR$(165)>]SCORE:    "Q
7040 PRINT"[<1CRSR U>][<20CRSR R>][<1CHR
$(167)>]"
7045 PRINT"[<1CHR$(204)>][<19CHR$(175)>]
[<1CHR$(186)>]"
7046 PRINT"[<1ARA>][<1CHR$(207)>][<19CHR
$(183)>][<1CHR$(208)>]"
7047 PRINT"[<1CHR$(165)>]HI-SCORE:"HI
7048 PRINT"[<1CRSR U>][<20CRSR R>][<1CHR
$(167)>]"
7049 PRINT"[<1CHR$(204)>][<19CHR$(175)>]
[<1CHR$(186)>]":PRINT"[<1CYN>][<4CRSR D>
]":GOSUB13000:OC=0:POKEW(2),0
7050 FORP=1TO7000:NEXT
7060 POKE2042,13:POKE2043,15
7070 RETURN
8000 POKEW(3),0:POKEFH(3),9:POKEFL(3),25
0:POKEHH(3),0*16+2:POKEW(3),129
8040 POKE2042,14:POKE2043,14:IFQ>RETHENR
E=RE+5000:I=I+1:GOSUB12500
8050 I=I-1:IFI=-1THEN10000
8060 FORP=200TO20STEP-20:POKEW(2),0:POKE
FH(2),P:POKEW(2),17:FORU=1TO100:NEXT
8070 NEXT:POKEW(2),0:POKEFH(2),69
8100 PRINT"[<1HOME>]
                    ";
8110 PRINT"
          ";:IFUD=-1THENSK$="SKULLS"
8115 PRINT"
          ":IFUD=+1THENSK$="TOMBS"
8120 PRINT"[<1HOME>][<1RVS>][<1PUR>]
  SCORE      ","LIFES",SK$

8130 PRINTQ,,I+1,50+AU-NS
8410 FORU=1TO3000:NEXT
8420 POKE2042,13:POKE2043,15:POKEFH(3),2
0:POKEHH(3),4*16+2
8500 GOTO90
10000 FORP=250TO10STEP-20:POKEW(3),0:POK
EFH(3),P:POKEFL(3),P:POKEW(3),33
10010 FORU=1TO50:NEXTU,P:POKEW(3),0:PRIN
T"[<1PUR>]"
10020 FORP=1TO9
10030 PRINT"[<1HOME>][<10CRSR D>][<15CRS
R R>]"MID$(GA$,1,P)
10040 FORU=1TO40:NEXTU,P
10045 PRINT"[<1HOME>][<1RVS>]SCORE"Q
10050 FORP=1TO3000:NEXT
10060 IFQ>HITHENHI=Q:GOSUB12000
10065 POKEFL(3),100:POKEFH(3),20:POKEHH(
3),4*16+2:POKEV+21,0
10070 GOTO80
11000 FORP=1TO3:POKEW(P),0:NEXT
11010 PRINT"[<1CLR>][<1CYN>]"
11020 FORP=1TO9:PRINTDI$(P,0),,DI$(P,2):
NEXT
11030 POKEV+6,0:POKEV+7,60:POKEV+8,0:POK
EV+9,50:POKEV+21,24
11040 FORP=0TO150:POKEV+6,P:POKEV+8,P+10
```

NUOVISSIMI
Per Commodore 64

Codice 3

Avete mai avuto la vita di un'altra persona tra le vostre mani? Cosa fareste se vi capitasse di ricevere una disperata richiesta di aiuto?
In questa situazione avrete bisogno di tutto il vostro sangue freddo, perché un uomo sta morendo, e voi siete gli unici che possono aiutarlo.
Alle vostre conoscenze mediche, ma soprattutto al vostro buonsenso, il compito di risolvere i guai, in lotta contro le difficoltà di comunicazione, la malattia e il tempo.
Estremo realismo per un gioco che è anche un modo per acquisire un'utilissima esperienza di pronto soccorso nonché controllo dei propri nervi.

Disco o cassetta **Lit. 30.000**

TERRAGON

Tattica e strategia tra le stelle, in una guerra che, per non distruggere la terra in un olocausto nucleare, diventa una corsa selvaggia all'espansione nello spazio.
Divenuti comandanti di una supernave del futuro dovrete esplorare e colonizzare lo spazio, tra pericoli di alieni buchi neri, pianeti fantasma, distorsioni magnetiche ed altri ancora, starà all'intelligenza con cui avrete costruito il calcolatore di bordo, oltre che alla vostra velocità di reazione ed abilità di manovra, il risultato finale.
Da uno a quattro giocatori per un gioco che unisce costruzione logica e possibilità tattiche ad un livello pari a quello degli scacchi 3-D con una elevata facilità di apprendimento.

Disco o cassetta **Lit. 30.000**

Come sempre disponibili:

— ALTO MEDIOEVO
Gioco di simulazione economica in un ambiente feudale.
Da 1 a 9 giocatori.

Disco o cassetta **Lit. 30.000**

— ATOMO
Una vera centrale atomica per il vostro C-64.

Disco o cassetta **Lit. 30.000**

— HANOI + OTHELLO
Disco o cassetta **Lit. 30.000**

— HIDDEN CODE + BIORITMI
Disco o cassetta **Lit. 30.000**

— BLACK JACK
Disco o cassetta **Lit. 30.000**

— DATA BA.SE SORG.
Programma sorgente per la creazione di archivi, usa files relativi con catalogo sequenziale.
Lunghezza e numero record definibili dall'utente.
In Basic non protetto con istruzioni.

Solo dischetto **Lit. 50.000**

ACQUISTIAMO PROGRAMMI ORIGINALI

Realizziamo programmi su commissione a seconda delle vostre esigenze su:

COMMODORE VIC 20
COMMODORE C-64
OLIVETTI M10
OLIVETTI M20
IBM PC

BA.SE s.n.c.

SOFTWARE HOUSE
Casella Postale 4-13055
Occhieppo Inferiore (VC)
Tel. 015/592730

Ricco package di programmi gestionali (fatturazione, magazzino, condominio ecc.)

A vostra disposizione per:

SOFTWARE APPLICATIVO
AUTOMAZIONE DI PROCESSO
SOLUZIONE DEI VOSTRI PROBLEMI
CORSI BASIC INDIVIDUALI E DI GRUPPO

Giochi per Texas TI/99

TILANDIA

Il re di Tilandia si deve assentare per quattro anni, e vi lascia in custodia il suo regno.
Ma, una volta alle prese con i regali grattacapi, scoprirete che fare il re non è poi tanto rilassante.
Attenti a guerre, amministrazione del territorio, e malcontento popolare, perché potrebbe succedervi di svegliarvi con un cappio per cravatta tra i contadini in rivolta.

Cassetta **Lit. 25.000**

SUPERBOWL

Tutti avranno prima o poi sognato di poter far parte di una squadra di football americano che sta per giocare alla finale del campionato.
Bene, il vostro Texas vi da ora la possibilità di essere il qarterback, il regista della squadra, senza rischiare neppure un graffio. Ma attenzione, è proprio lui a giocare per la squadra avversaria.

Cassetta **Lit. 25.000**

LABIRINTO 3-D

Un labirinto tridimensionale da cui cercare di uscire il più rapidamente possibile, perché all'uscita...

Cassetta **Lit. 25.000**

SPACE TI/99

Vieni eletto comandante della spazionave Texas, ma devi prima trovare, e poi distruggere, le navi nemiche lungo una scacchiera di 64 costellazioni, usando tutta la tua intelligenza per organizzare la caccia in modo da sfruttare al meglio la tua limitata autonomia.

Cassetta **Lit. 25.000**

POKER + ROULETTE

Combinazione di due giochi più che classici in cui il calcolatore svolge il ruolo di croupier.

Cassetta **Lit. 30.000**

NUOVISSIMO PER ATARI 800 - 600 XL - 800 XL

ALTO MEDIOEVO

Finalmente anche per Atari il gioco che ha fatto impazzire i commodoriani...
In tutto e per tutto identico alla versione C-64.

Cassetta **Lit. 30.000**

Spedire in busta chiusa a:

BA.SE s.n.c. - Casella Postale 4
13055 Occhieppo inf. (VC)

Nome e Cognome ____________________

Indirizzo ____________________

Cap. ________ Città ________ Provincia ________

VIC 20 ☐ C 64 ☐ IBM PC ☐ M10 ☐ M20 ☐ Atari ☐ Texas ☐

Ordino n° ________ ☐ Disco ☐ Cassetta ________

Ordino n° ________ ☐ Disco ☐ Cassetta ________

Per un totale di Lire ____________________

Pagamento ☐ Allegato assegno non trasf. sped. celere ☐ Contro assegno + spese postali

Death per C 64

Seguito listato Death.

```
:NEXT
11050 FORP=60TO200:POKEV+7,P:POKEV+9,P-1
0:NEXT
11060 PRINT"[<1HOME>][<21CRSR D>]"TAB(10
)"[<1ARA>]BONUS [<1R.C.>]1000 [<1ARA>]PO
INTS":Q=Q+1000
11070 FORP=1TO9:POKEW(3),0:POKEFH(3),N1(
P):POKEFL(3),N2(P):POKEW(3),33
11080 FORU=1TODU(P):NEXT:NEXT
11085 POKEHH(3),4*16+2
11090 UD=-UD:SG=SG+1:AU=AU+10:NT=NT-2:IF
NT<8THENNT=12
11095 IFUD=1ANDNT=8THENNT=9
11100 PRINT"[<1CLR>]":NS=0:POKEV+21,4:PO
KEW(3),0:POKEV+7,140:GOTO85
12000 PRINT"[<1CLR>]WHAT'S YOUR NAME?[<2
CRSR D>]":HI$="":TM=0:LT=0:ID=0:POKEV+21
,0:POKE198,0
12010 GETA$
12020 TM=TM+1:IFTM=4000THENRETURN
12030 IFA$=""THEN12010
12035 IFASC(A$)=13THENRETURN
12040 IFASC(A$)=20THENID=ID+1:IFID<=LTTH
ENPRINTA$;:HI$=HI$+A$:GOTO12010
12050 IFASC(A$)=148ANDID>LTTHEN12010
12055 IFA$=" "THENHI$=HI$+A$:PRINTA$;:LT
=LT+1:IFLT=15THENRETURN
12057 IFA$=" "THEN12010
12060 IFASC(A$)<65ORASC(A$)>90THEN12010
12070 LT=LT+1:HI$=HI$+A$:PRINTA$;:IFLT=1
5THENRETURN
12080 GOTO12010
12500 POKE53280,14::FORU=1TO300:NEXT:POK
E53280,0:RETURN
13000 IFOC=1THEN13010
13005 PRINT"[<1HOME>][<1RVS>]HI-SCORE:"
HI,HI$
13010 PRINT"[<5CRSR D>]"
13020 PRINT"[<1RVS>]    [<1CHR$(223)>][<
1CRSR R>]    [<1CRSR R>][<1CHR$(169)>]
[<1CHR$(223)>][<1CRSR R>]      [<1CRSR R>]
 [<2CRSR R>] "
13030 PRINT"[<1RVS>] [<2CRSR R>] [<1CRS
R R>] [<4CRSR R>] [<2CRSR R>] [<2CRSR R>
]![<1RVS OFF>]![<1RVS>][<2CRSR R>] [<2
CRSR R>] "
13040 PRINT"[<1RVS>] [<2CRSR R>] [<1CRS
R R>]    [<2CRSR R>] [<2CRSR R>] [<2CRSR
R>]![<1RVS OFF>]![<1RVS>][<2CRSR R>]
 "
13050 PRINT"[<1RVS>] [<2CRSR R>] [<1CRS
R R>] [<4CRSR R>]     [<2CRSR R>]![<1RVS
 OFF>]![<1RVS>][<2CRSR R>] [<2CRSR R>]
"
13060 PRINT"[<1RVS>]    [<1RVS OFF>][<1
CHR$(169)>][<1RVS>][<1CRSR R>]     [<1CR
SR R>] [<2CRSR R>] [<2CRSR R>]![<1RVS O
FF>]![<1RVS>][<2CRSR R>] [<2CRSR R>] "
13070 RETURN
14000 RN=INT(RND(1)*15+1)
14010 ONRNGOSUB30000,30010,30020,30030,3
0040,30050,30060,30070,30080,30090,30100
14020 IFRN<12THENRETURN
14030 ONRN-11GOSUB30110,30120,30130,3014
0
14040 RETURN
15000 GETA$:IFA$<>""THEN84
15010 GOTO85
30000 PRINT"[<1WHT>]";:RETURN
30010 PRINT"[<1RED>]";:RETURN
30020 PRINT"[<1CYN>]";:RETURN
30030 PRINT"[<1PUR>]";:RETURN
30040 PRINT"[<1GRN>]";:RETURN
30050 PRINT"[<1BLU>]";:RETURN
30060 PRINT"[<1YEL>]";:RETURN
30070 PRINT"[<1ARA>]";:RETURN
30080 PRINT"[<1MAR>]";:RETURN
30090 PRINT"[<1R.C.>]";:RETURN
30100 PRINT"[<1G.1>]";:RETURN
30110 PRINT"[<1G.2>]";:RETURN
30120 PRINT"[<1V.C>]";:RETURN
30130 PRINT"[<1AZZ>]";:RETURN
30140 PRINT"[<1G.3>]";:RETURN
```

I tasti grafici e speciali sono stati racchiusi tra i simboli < >, all'interno di parentesi quadre e preceduti da un numero che specifica quante volte occorre digitare quel determinato carattere. Es. [<8CRSR D>], digitare per 8 volte il tasto di spostamento verso il basso del cursore. Per i caratteri grafici viene indicato il codice ASCII corrispondente; consultare il manuale per la conversione.

alla linea 10000, dove viene determinata la fine della partita.

Le linee da 10000 a 10070 determinano la fine della partita e, se il punteggio raggiunto è superiore al record registrato nella variabile HI, mandano alla routine 12000 - 12080.

La linea 10070 manda il programma allo schermo di presentazione, per iniziare una nuova partita.

Le 11000 - 11100 vengono eseguite alla fine di ogni livello, e provvedono a visualizzare il bonus di 1000 punti, ad aggiornare il punteggio e ad aumentare la difficoltà per il livello successivo. Le linee 11070 e 11080 suonano una musichetta che sottolinea il bonus.

La subroutine 12000 - 12080 aggiorna le variabili HI e HI$, dove sono registrati il massimo punteggio e il nome del giocatore che l'ha realizzato.

La linea 12500 colora di blu per pochi attimi il bordo dello schermo: questo significa che si è guadagnata una vita extra.

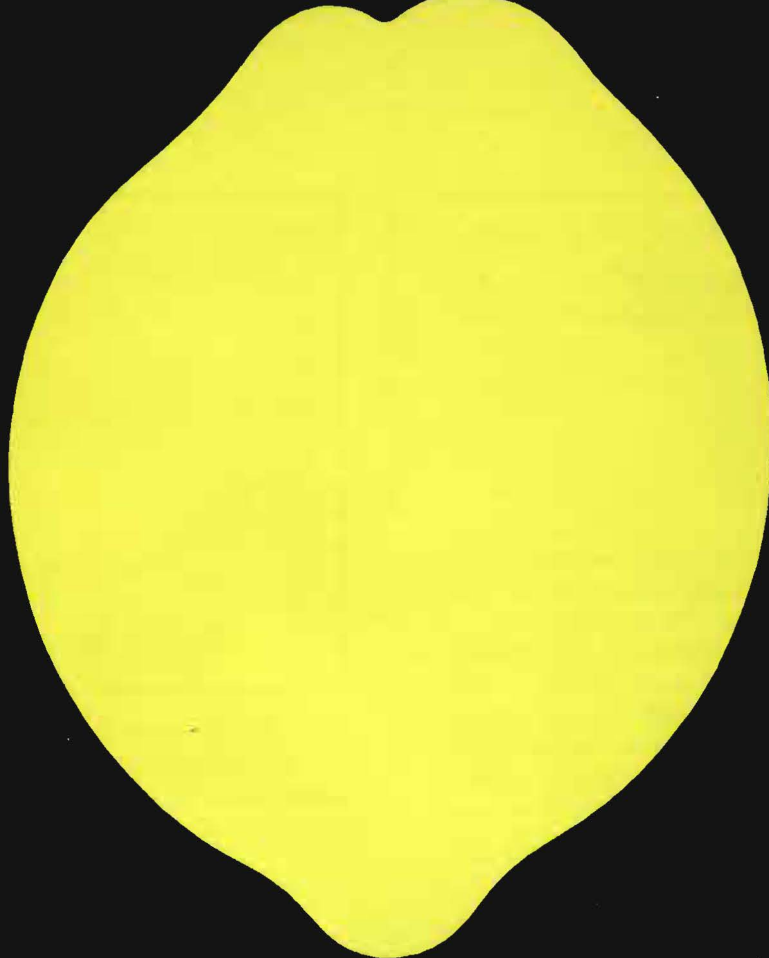
Le linee comprese fra 13000 e 13070 creano lo schermo di presentazione con la scritta Death.

Le linee 14000 e 14040 generano un numero casuale in base al quale viene modificato il colore della scritta Death nello schermo di presentazione.

Le linee 15000 e 15010 bloccano l'inizio del gioco se non si preme il pulsante "F1".

Le linee 30000 - 30140 contengono 15 subroutine in ognuna delle quale è memorizzato uno dei 15 colori disponibili sul Commodore 64. Queste subroutine vengono utilizzate per la colorazione dei teschi, delle tombe e della scritta Death. ■

Per la sete di soft



I personal computer della serie LEMON II, JEN PC 1, PC 2 e JEN PC 3 Biprocessore, sono grandi compatibili.
L'accesso a tutto il soft CP/M, 20.000 programmi, è la loro caratteristica più evidente.
Ma sono garanzia, assistenza e un prezzo conveniente a fare di LEMON II una serie di computer realmente vincenti.



MODELLO	JEN PC 1	JEN PC 2	JEN PC 3
PROCESSORE	6502	6502	6502/Z80
RAM	48K	64K	64K
EPROM INTERPRETE	10K	10K	10K
EPROM MONITOR	2K	2K	2K
SISTEMA OPERATIVO DOS	S	S	S
SISTEMA OPERATIVO CP/M	E	E	S

LEMON II
il grande compatibile

Distribuito dalla Belton Electronics s.r.l. - Zona Industriale 62010 Montelupone (MC) - Italia - Tel. 0733/586423-24 Telex 600071 JENSPA I - Divisione Informatica della Jen Elettronica s.r.l.

Zeta2

Indagine su alcune strane figure

Siete sicuri di saper rappresentare il grafico di una funzione?

di *Giulio Morpurgo*

Introduzione

Scopo di questo articolo è mostrare alcune figure, a prima vista inattese, che si ottengono eseguendo un semplice programma, e spiegare i motivi della loro comparsa. Poche formule matematiche e alcuni facili concetti intuitivi sono sufficienti a giustificare qualitativamente e quantitativamente le caratteristiche di queste strane figure. La conoscenza di questi concetti è utile per chiunque debba rappresentare grafici di funzioni con il calcolatore.

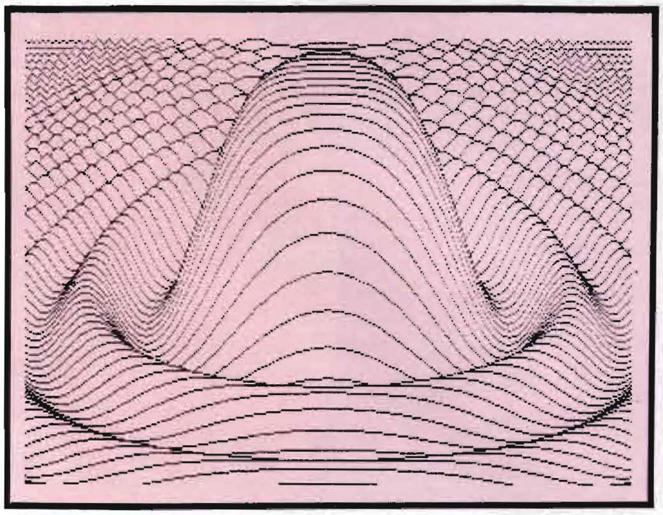


Figura 1. *Grafico tridimensionale della funzione sin $(r^2)/r^2$. Si può notare come le "gobbe" nella parte alta della figura sembrino essere disposte circolarmente intorno ad un punto più in alto del margine del disegno.*

Analisi delle simmetrie

Tutto cominciò quando, facendo girare un mio programma per grafici tridimensionali, ottenni la figura 1. La funzione della quale volevo disegnare il grafico era $z = \sin(r^2)/r^2$ (dove $r^2 = x^2 + y^2$). Questa funzione, poiché r è l'unica variabile, ha una simmetria circolare intorno all'origine delle coordinate x e y, che nel grafico si trova al centro della figura. Simmetria circolare vuol dire che, se si traccia una circonferenza di raggio R qualsiasi con centro nell'origine, su tutti i punti della circonferenza la funzione ha lo stesso valore.

Sul grafico ottenuto dal programma si può osservare molto bene la simmetria circolare nella zona centrale della figura, mentre si vede che ai bordi questa simmetria diventa confusa e non è più riconoscibile. Sembra quasi che le piccole gobbe, che si vedono chiaramente, prendano il sopravvento su di essa, e inoltre che queste gobbe tendano a disporsi circolarmente intorno a un punto (in alto, fuori dallo schermo) diverso dall'origine. La situazione è simile a quella della figura 3b, nella quale si vede come la struttura ad anelli circolari concentrici intorno al centro della figura manifesti delle irregolarità verso i bordi del disegno.

Come mai la simmetria circolare è in parte scomparsa? Dapprima pensai ad una illusione ottica; poi, riflettendo un momento, cominciai a sospettare quale fosse il motivo di questa inattesa rottura di simmetria; questo sarà l'oggetto della discussione seguente.

Per prima cosa, conviene cercare di rappresentare meglio tutti i particolari della situazione. Mi limiterò a considerare la "proiezione" della funzione sin (r^b) sul piano x, y, rappresentandone il segno con un puntino nero se positivo, bianco se negativo; per fare ciò userò il programma del listato 1, il quale, definita una funzione z all'interno della subroutine 1000, consente di rappresentare il segno su un settore del piano xy, delimitato dalle rette $x = XM$, $x = -XM$, $y = YM$, $y = -YM$ (figura 2a), dove i valori di XM e di YM sono ogni volta scelti e dati in input al programma. Com'è ovvio, ogni funzione può essere calcolata solo in un numero finito di punti: questi punti sono rappresentati dalle intersezioni delle linee tratteggiate orizzontali e verticali nel reticolo di figura 2b, che è solo una porzione del reticolo completo 216 x 216. Il valore della funzione verrà dunque calcolato esclusivamente in questi punti; e, come ripeto, ciascuno dei 216 x 216 puntini visualizzati sullo schermo è colorato in nero se il valore della funzione in quel punto è maggiore o uguale a 0, è invece bianco se la funzione è

negativa.
La distanza tra un punto del reticolo ed il successivo vale $\Delta x = 215/2XM$ nella direzione dell'asse x, e $\Delta y = 215/2YM$ nella direzione dell'asse y. Per comodità, d'ora in poi sarà sempre XM = YM, e quindi $\Delta x = \Delta y$.
Per semplificare una varietà di situazioni, riporto qui di seguito le figure 3a-3p, ottenute usando la funzione $z = \sin(r^b)$ e variando sia l'esponente b che il parametro XM.
La figura 3a è un buon esempio di quello che ci si aspetterebbe veder comparire sul video; la proiezione di una funzione oscillante in segno in dipendenza della sola variabile r, e con periodo di oscillazione sempre più corto al crescere di r.
Osservando però le altre figure, soprattutto dalla figura 3 c in avanti, ci si accorge che sul video avvengono cose stranissime, totalmente diverse da quello che ci si aspettava. È come se, invece della funzione $\sin(r^b)$, il programma ne usasse un'altra, completamente diversa.
Dall'osservazione di queste figure si nota come la simmetria circolare, e cioè la struttura ad anelli concentrici attorno al centro delle figure, è perduta mano a mano che ci si allontana dall'origine. Ciò è tanto più evidente quanto più è grande XM (e, in ultima analisi, quanto più grande è Δx, il passo del reticolo di figura 2b).
Al posto della vecchia simmetria intervengono altri tipi di simmetrie e di regolarità, che adesso analizzerò in dettaglio.
Tre cose balzano subito evidenti agli occhi:
1) in ognuna di queste figure, anche se la simmetria circolare è andata perduta, è rimasta una simmetria tale che, invertendo la figura rispetto a uno degli assi o ad una diagonale, la figura non cambia;
2) oltre alla simmetria circolare che si può osservare al centro di ogni figura, vi sono, in alcune di queste figure, altre zone localmente simmetriche distribuite con una certa regolarità. Queste zone hanno dimensioni uguali a quella centrale quando $b = 2$, minori quando $b > 2$, e maggiori quando $b < 2$;
3) in alcune di queste figure, tra le varie zone a simmetria circolare bene evidenti, se ne notano altre meno definite, e si può osservare come anche queste abbiano una distribuzione regolare.

Mi propongo ora di dare una spiegazione a tutte queste apparenti stranezze. Per far questo cercherò di chiarire successivamente i seguenti punti:
1) come mai la simmetria circolare è rotta in modo

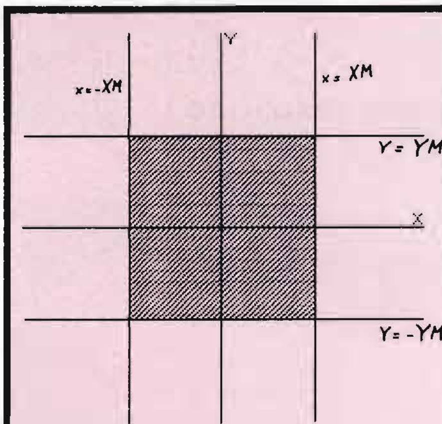


Figura 2a. *La parte tratteggiata del piano x, y rappresenta il campo sul quale viene rappresentato il segno della funzione data.*

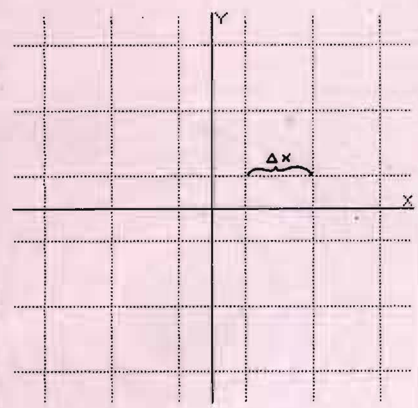


Figura 2b. *I punti di intersezione delle linee orizzontali e verticali del reticolo tratteggiato sono i punti per i quali il programma calcola il valore della funzione. Δ x è la variazione in x al passare da un punto del reticolo al successivo, e vale Δ x = 2 ★ XM/215.*

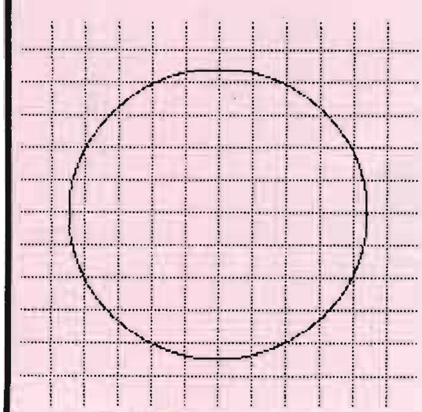


Figura 2c. *I punti del reticolo che corrispondono ad un dato valore di r sono pochi.*

Indagine su alcune strane figure

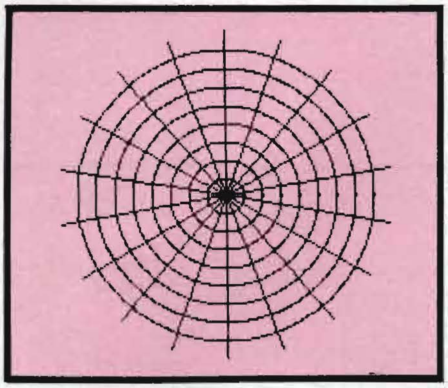
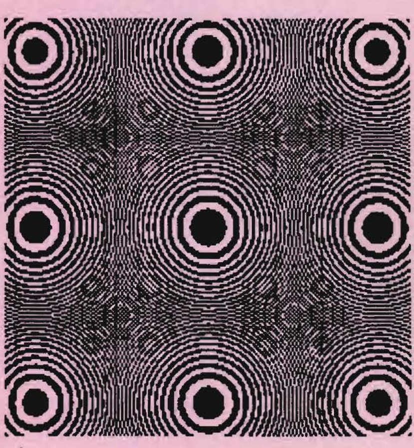
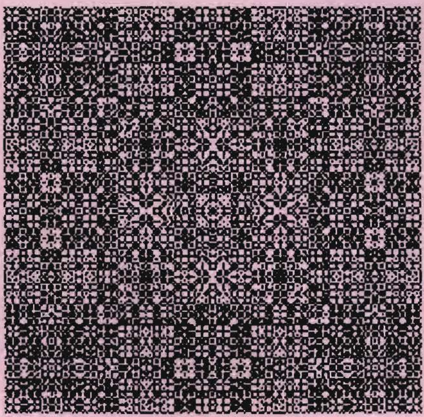


Figura 2d. *Un reticolo di questo tipo sarebbe più adatto per rappresentare una funzione a simmetria circolare.*

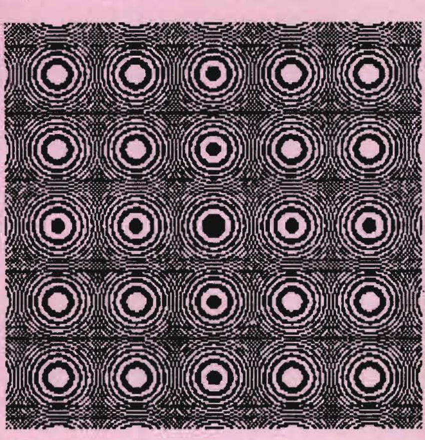
Figura 3a-p. *Le figure ottenute facendo girare il programma 1 con diversi valori di b e di XM.*



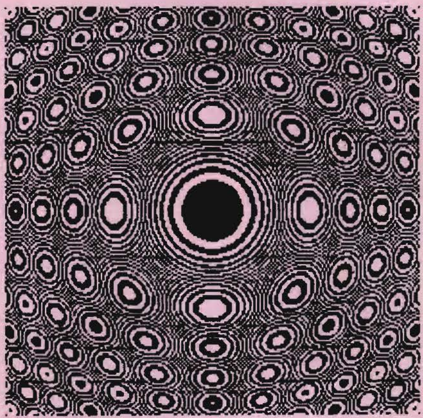
c)



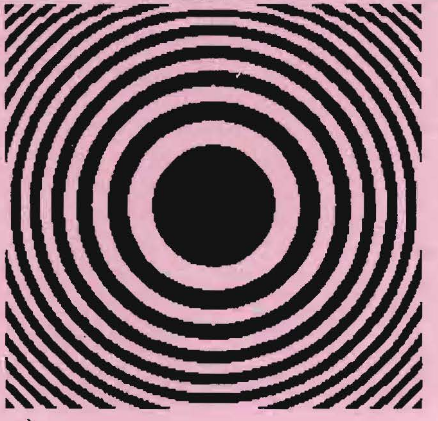
g)



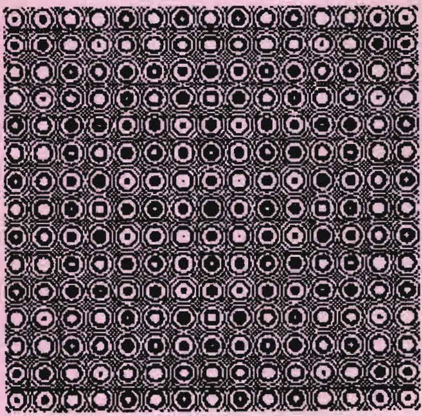
d)



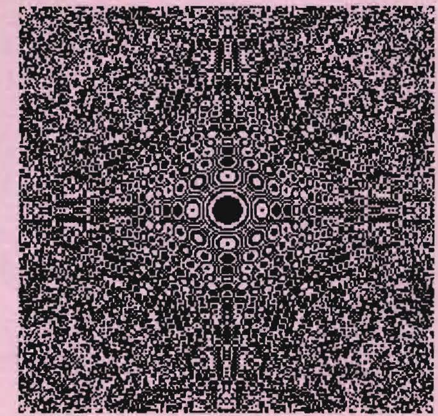
h)



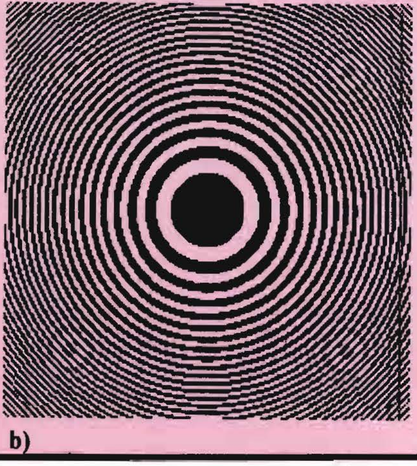
a)



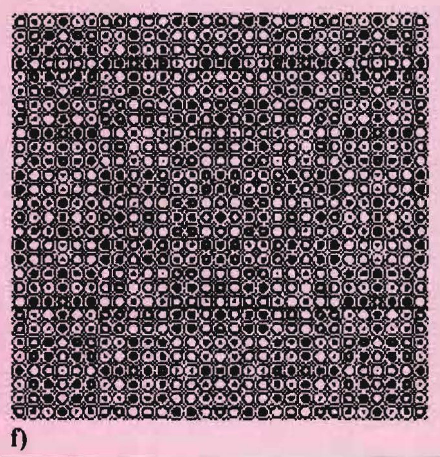
e)



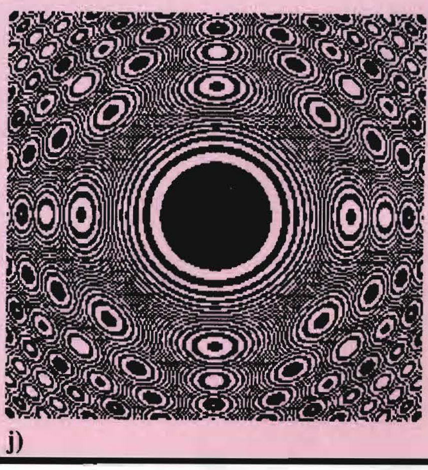
i)



b)



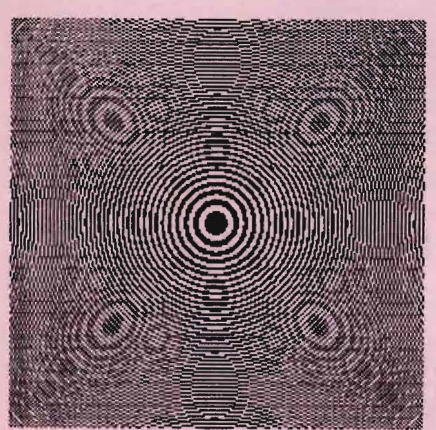
f)

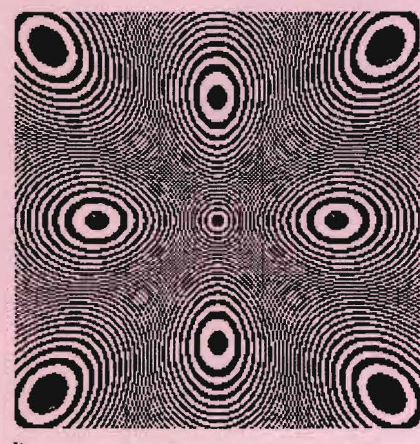


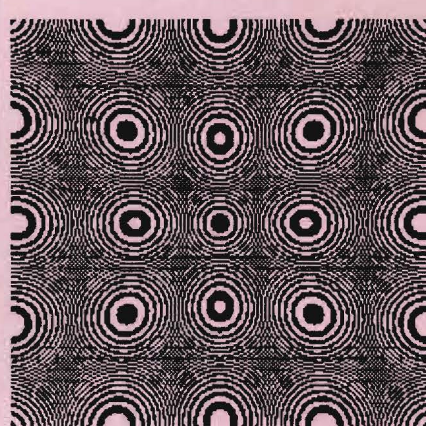
j)

PERSONAL SOFTWARE

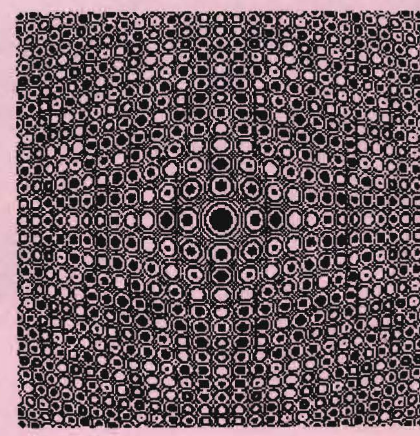
Indagine su alcune strane figure

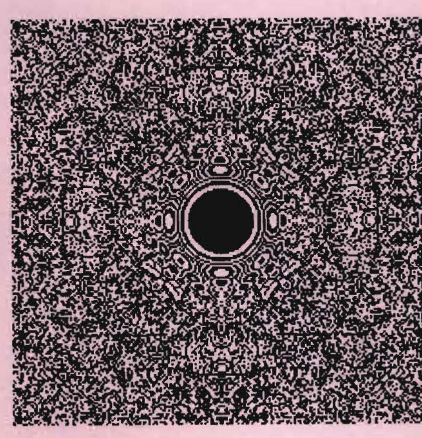

k)


l)


m)


o)


n)


p)

Seguito figura 3.

così vistoso;
2) come mai rimane una simmetria quadrata;
3) qual'è la ragione per cui si hanno, in parecchi casi, simmetrie circolari centrate in punti diversi dall'origine.

Perché la simmetria circolare sparisce

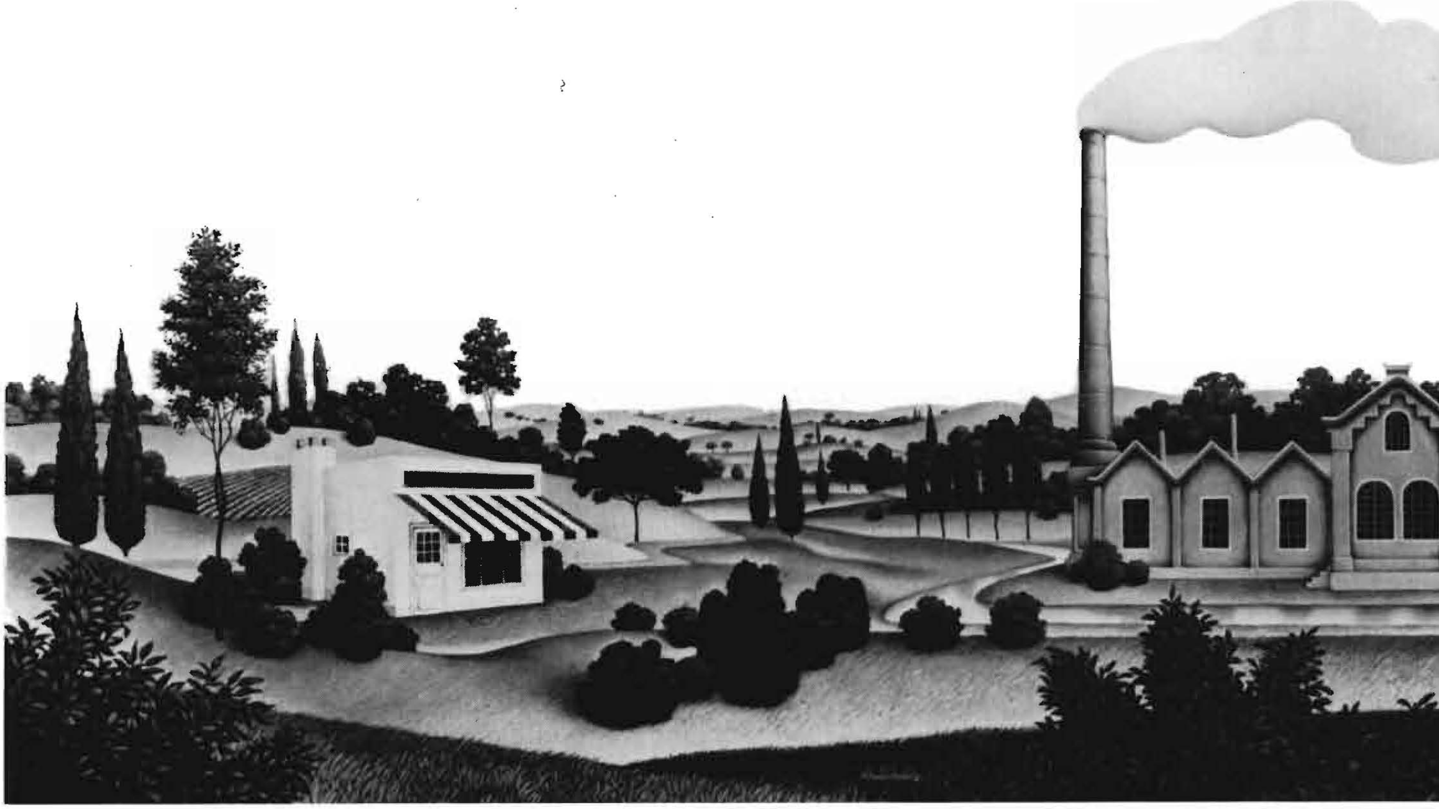
Il motivo di questa inattesa e impressionante rottura di simmetria è legato a due fatti:
1) il reticolo usato per rappresentare la funzione ha simmetria quadrata e non circolare e inoltre esso è discreto e non continuo (come già detto, la funzione è calcolata solo in 216 x 216 punti);
2) abbiamo a che fare con una funzione (sin r^b) che, quando b è maggiore di 1, al crescere di r oscilla molto rapidamente in segno (cioè, preso un valore di r sufficientemente grande, l'intervallo Δ r, tale che il segno della funzione calcolato in r e quello calcolato in r + Δ r siano opposti, diventa piccolo a piacere). Nel seguito della discussione chiarirò come questi due fatti provochino la stranezza delle figure.
È conveniente partire da una osservazione; nelle figure (ad esempio 3c) la simmetria circolare è data da una serie di anelli concentrici di spessore sempre minore via via che ci si allontana dall'origine. Ad una certa distanza dall'origine lo spessore di questi anelli è di un solo puntino, mentre oltrepassata questa distanza la struttura ad anelli scompare (o meglio, ne compare un'altra centrata in un altro punto) e la simmetria circolare si rompe.
È ovvio che non è possibile continuare a fare gli anelli sempre più sottili, perché la risoluzione del video (e del nostro reticolo, e questo è il punto essenziale) è discreta e non continua; la domanda che ci si può allora porre è questa: in che modo si dovrebbe manifestare la simmetria circolare oltrepassata questa distanza dall'origine?

Per rispondere a questa domanda, ricordando che le funzioni dalle quali derivano le figure 3 sono del tipo z = sin (t), con t = r^b, analizzeremo in dettaglio cosa succede quando si rappresenta la funzione sin (t) calcolandone il valore in un numero finito di punti. Per fare ciò, si scelgono un certo numero di valori di t, che per comodità terremo equidistanti tra loro, e si calcola il valore della funzione in corrispondenza dei valori di t scelti.
Dalle figure 6 si nota come, fino a che noi "campio-

"Cara IBM, in un'attività come la mia è veramente utile il tuo Personal Computer?"



Si! Si!

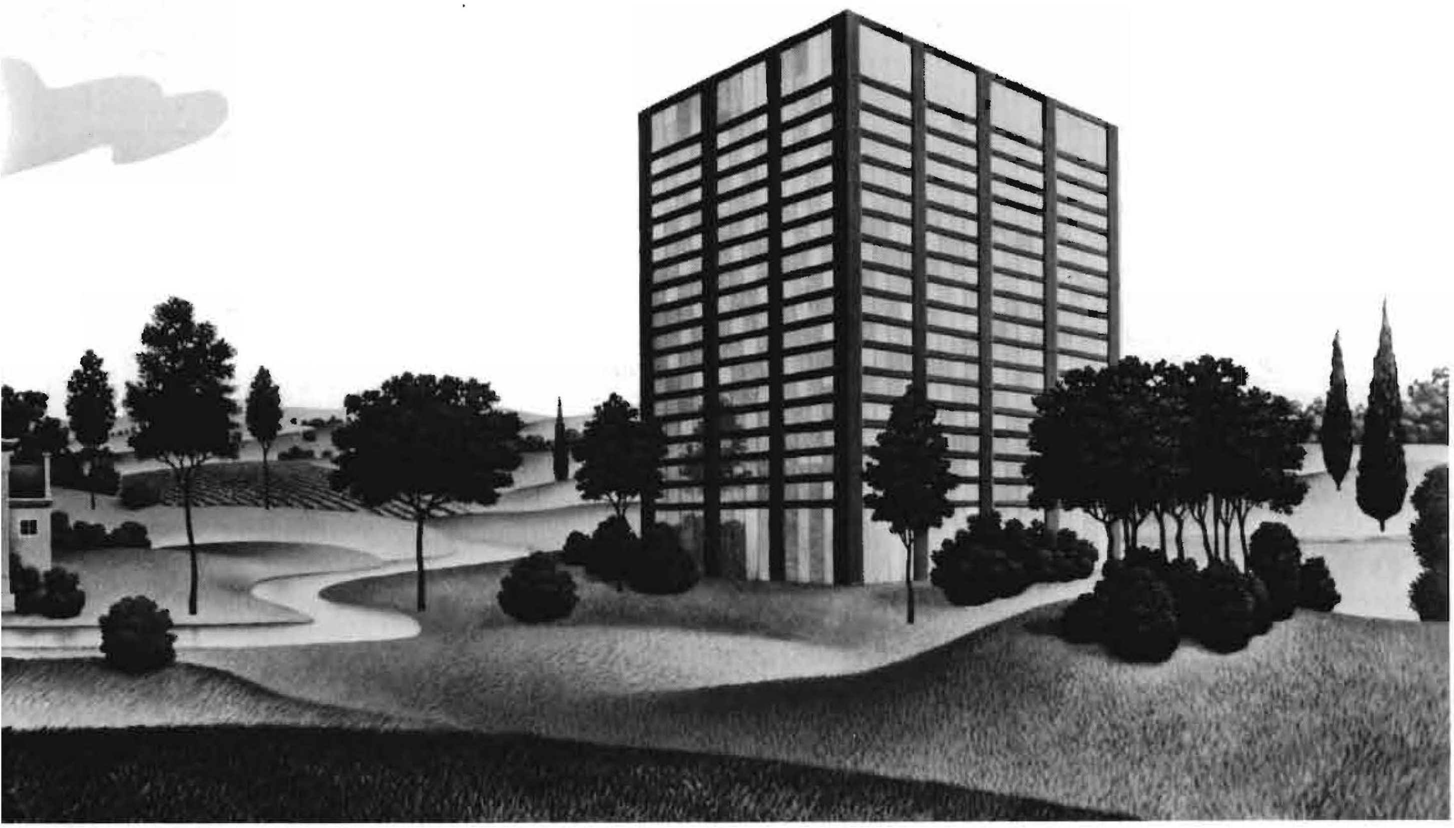


Il Personal Computer IBM è utile in ogni caso. A partire proprio dalle attività che richiedono un organico minimo, anche di una persona sola. E' il tuo caso? Allora puoi usare il Personal Computer IBM per ottimizzare il tuo personale rendimento professionale, per rendere più veloce ed efficiente la gestione della contabilità e per tenere sempre in perfetto equilibrio le entrate e le uscite. Ci sono centinaia di programmi già pronti che risolvono qualunque problema. Anche il tuo.

Hai invece una media azienda? Le cose non cambiano. Col Personal Computer IBM e la sua unità di espansione, puoi valutare, pianificare e controllare il lavoro, puoi vendere e comprare più efficacemente e approfittare tempestivamente delle occasioni (con la sicurezza che i tuoi dati sono sempre perfettamente protetti).

E per finire, c'è un Personal Computer IBM anche per la grande azienda: il modello XT 370, una vera e propria "stazione di lavoro" indipendente, che puoi anche collegare all'elaboratore centrale. Per creare una maggiore comunicazione tra i repar-

Si!



ti e per aiutare i manager a prendere sempre le decisioni migliori.

Qualunque sia il tuo caso, i concessionari IBM ti stanno aspettando per mostrarti quanti vantaggi puoi avere dal Personal Computer IBM, per consigliarti sugli acquisti e per offrirti un servizio e un'assistenza sempre impeccabili.

Cerca il concessionario IBM più vicino a te, nell'elenco della pagina seguente e vai a trovarlo.



Ti mostrerà tutta la grande versatilità del Personal Computer IBM.

Per ricevere maggiori informazioni spedisci questo coupon a:
IBM Italia Distribuzione Prodotti S.p.A. - Milano 2
Palazzo Cellini - 20090 Segrate Milano

Nome .... PS

Azienda ....

Indirizzo ....

Telefono ....

IBM

Indagine su alcune strane figure

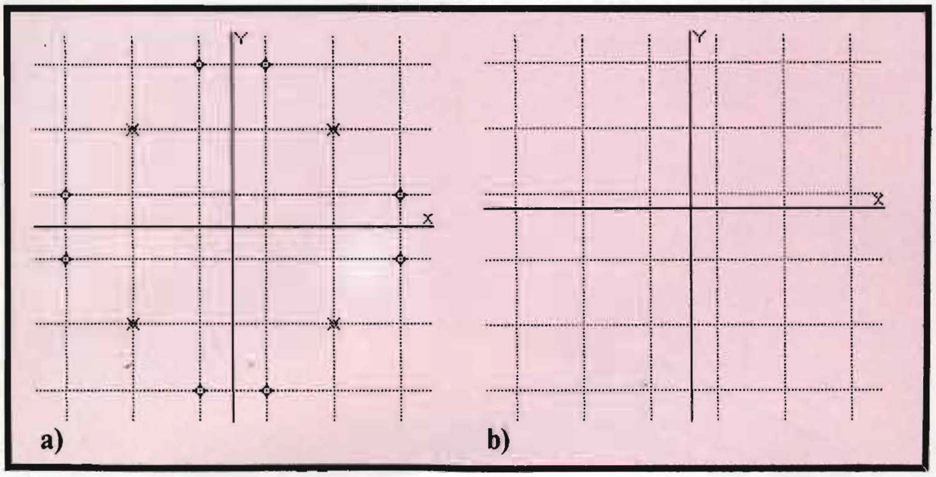


Figura 4. *Il reticolo che definisce i punti per i quali il valore della funzione viene calcolato è centrato intorno all'origine (figura 4a). Ciò fa si che, dato un punto del reticolo, ne esistano anche altri (3 oppure 7) con lo stesso valore di r. Poiché la funzione scelta dipende solo da r, si ha così la simmetria quadrata descritta nel testo.*
Se decentriamo il reticolo, ad esempio spostando l'origine delle coordinate, questa corrispondenza si perde (figura 4b) e si perde anche la simmetria quadrata (figura 5).

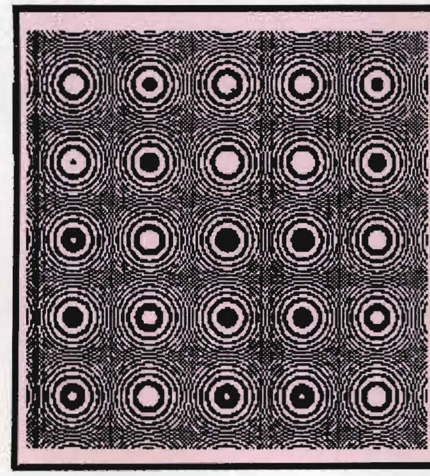


Figura 5. *La stessa figura di 3d, ma questa volta il reticolo non è centrato nell'origine.*
Si può notare come sia andata perduta la simmetria quadrata.

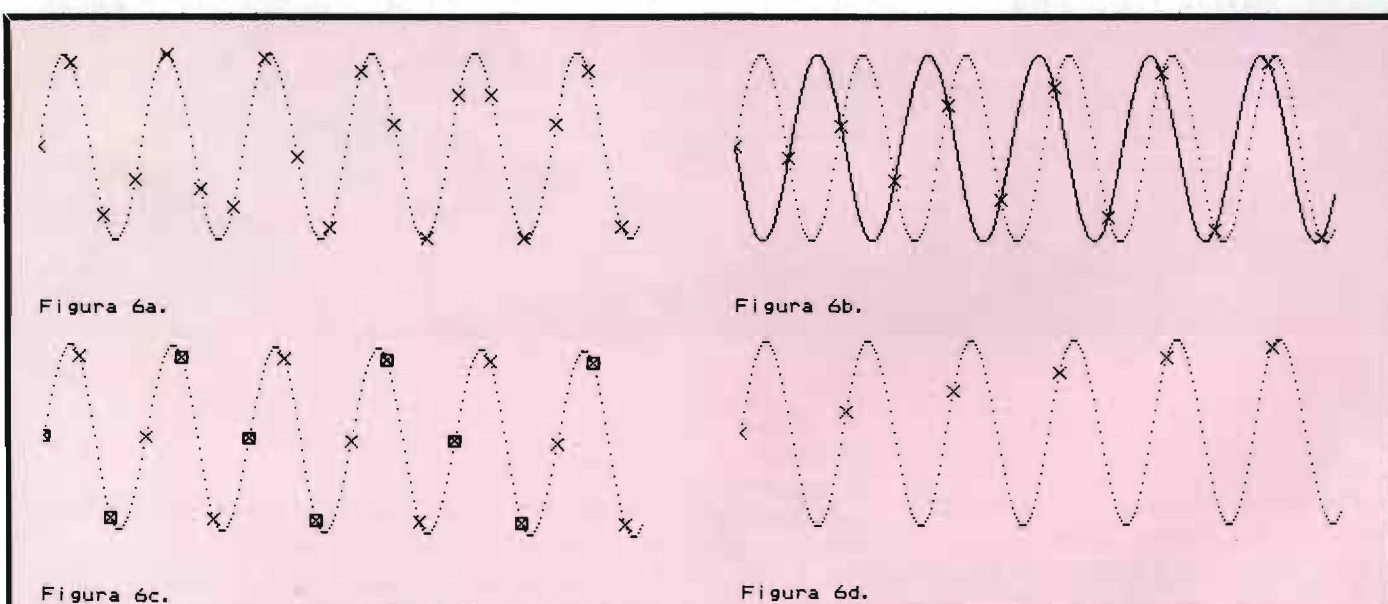


Figura 6. *Fino a che la distanza tra i punti che prendiamo per rappresentare la sinusoide è minore di π (6a), questa può essere ricostruita senza ambiguità a partire da essi. Ma quando la distanza è anche poco maggiore di π, possiamo trovare un'altra sinusoide, con frequenza più bassa della prima, che passa per i punti dati; quindi essi non rappresentano bene la sinusoide originale.*
Si noti come nel caso in cui la distanza è vicina a π (6b), si abbia una alternanza di punti positivi e negativi, nel caso in cui essa vale 120 o 240 gradi (figura 6c., punti rispettivamente marcati con la croce o con il quadrato) si abbiano due punti di un segno seguiti da uno di segno opposto, e infine, quando la distanza vale poco più di 2 π (6d), si abbia una lunga serie di punti dello stesso segno.

"Ed ecco chi mi garantirà un'assistenza qualificata per il Personal Computer IBM."



Il concessionario IBM. Un "vero esperto" di elaborazione dati che ti aiuterà a ottenere il massimo dal tuo Personal Computer IBM e saprà garantirti un'assistenza puntuale e un servizio efficiente e affidabile. Insomma, un servizio all'altezza del nome IBM.

ABRUZZI/MOLISE
Campobasso - PUBLISISTEMI SRL - Via S. Antonio Abate, 236 - Tel. 0874.98144
Pescara - ITALDATA SRL - Via Tiburtina, 75 - Tel. 085.50843
Vasto - DATAGRAPH SRL - Corso Europa, 22 - Tel. 0873.535

BASILICATA
Matera - I.P.E.S. SPA - Via Annunziatella, 25 - Tel. 0835.216742
Potenza - I.P.E.S. SPA - Via Sanremo, 79 - Tel. 0971.43293

CALABRIA
Catanzaro - VISICOM SRL - Via XX Settembre, 62A/B/C - Tel. 0961.24181
Cosenza - CALIÒ SRL - Via N. Serra, 90 - Tel. 0984.32807
Reggio Calabria - SO.F.IN. SPA - Via S. Francesco da Paola, 108/D Tel. 0965.331648

CAMPANIA
Cava dei Tirreni - METELLIANA SPA - Via Mandoli, 16 - Tel. 089.463877
Napoli
ENGINEERING INFORMATICA SRL - Via Carducci, 15 - Tel. 081.402660
INFORMATICA CAMPANIA SPA - Via Orazio, 6/bis
INFORMATICA MERID. SNC - Via P. Castellino, 179 - Tel. 081.464022
POINTER SISTEMI SRL - Via A. De Gasperi, 45 - Tel. 081.312312
Salerno - OMNIA SRL - C.so Garibaldi, 47 - Tel. 089.220366
S. Maria Capuavetere - GENERAL SYSTEMS SRL - Via Unità d'Italia, 21/23 - Tel. 0823.811100

EMILIA
Bologna
ABACO INFORMATICA SAS - Via Bernini, 1 - Tel. 051.393274
CMB INFORMATICA SCRL - Via Arcoveggio, 74/10 - Tel. 051.323594
LUCKY SYSTEMS SRL - Via Farini, 33/A - Tel. 051.231569
STUDIO "P" COMM. SRL - Via Massarenti, 50 - Tel. 051.397660
SYSDATA ITALIA SPA - Via Massimo d'Azeglio, 58 - Tel. 051.330021
Carpi
DATA SRL - Via B. Peruzzi, 12 - Tel. 059.688090
UNIDATAX SRL - Viale N. Biondo, 6 - Tel. 059.698355
Faenza - DATA SERVICE SRL - Via Laderchi, 2 - Tel. 0546.660300
Ferrara - MARKITALIA COMPUTERS SRL - Via Bologna, 84 - Tel. 0532.96128
Forlì
C.E.DA.F. COOP. ELAB. DATI - Via Zanchini, 57 - Tel. 0543.65402
I.C.O.T. IMPIANTI SRL - Via Codazzi, 10 - Tel. 0543.724246
Imola - PALAZZO DONATO - Via Emilia, 23/A - Tel. 0542.29195
Modena
INTELCOM SRL - Via della Cittadella, 51/63 - Tel. 059.223663
MASETTI ELETTRONICA SRL - Corso Canalgrande, 14 - Tel. 059.219801
Parma
DS DATA SYSTEMS SRL - Borgo Lalatta, 8 - Tel. 0521.208193
PROGRAMMA NORD B SRL - Viale Mentana, 104 - Tel. 0521.96960
Piacenza - RCM COMPUTER SAS - C.so Vittorio Emanuele II, 96 - Tel. 0523.37656
Ravenna - CELCOOP SCRL - Via Faentina, 7 S. Biagio - Tel. 0544.462333
Reggio Emilia
ABAX INFORMATICA SCRL - Via M.K. Gandhi, 1/D - Tel. 0522.26941
A.P.E.D. ELABORAZIONE DATI - Via Filippo Re, 17 - Tel. 0522.38721
MEMAR ELECTRONIC SRL - Via M. Melato, 13 - Tel. 0522.94230
Rimini
HARD & SOFT SYSTEMS SRL - Viale Valturio, 43 - Tel. 0541.773343
TRE EMME PI SPA - Via P. Veronese, 14/16 - Tel. 0541.775153

LAZIO
Frosinone - SAIU ELETTRONICA SRL - Via Vado del Tufo, 85 - Tel. 0775.83093
Roma
BIT COMPUTERS SRL - Via F. Satolli, 35/57/59 - Tel. 06.6386146
CERVED SPA - Via Appia Nuova, 696 - Tel. 06.7940241
CRAMER ITALIA SPA - Via C. Colombo, 134 - Tel. 06.517981
DATAOFFICE SPA - Via Sicilia, 205 - Tel. 06.4754568
ELEDRA 3S SPA - Via G. Valmarana, 63 - Tel. 06.8127324
EXPO SAS - Via 4 Novembre, 151 - Tel. 06.6794293
GEDIN SRL - L.go D. De Dominicis, 7 - Tel. 06.432183
I.S.E.D. SPA - Via Tiburtina, 1236 - Tel. 06.4125851
ISI ITAL SISTEMI PER L'INFORMATICA SPA - P.zza SS Apostoli, 66 - Tel. 06.6793477
ITALSIEL SPA - Via Tevere, 26 - Tel. 06.84311
MEMORY COMPUTERS SRL - Via Aureliana, 39 - Tel. 06.4758366
MICROCOMP SPA - V.le M. Gelsomini, 28/30 - Tel. 06.5759324
NICA DIFF INF SRL - V.le Parioli, 40 - Tel. 06.872603
SAPES SRL - V.le Tito Livio, 12 - Tel. 06.3453536
SYS DATA ITALIA SPA - Via Cola di Rienzo, 265 - Tel. 06.351417
TELESIA SPA - Via V. Brancati, 64 - Tel. 06.5011551
VALDE ADEL SRL - P.zza S. Anastasia, 3 - Tel. 06.6786663
Viterbo - ITALBYTE SRL - V.le Trento - Pal. Garbini - Tel. 0761.221333

LIGURIA
Genova
DIFF.EL. SRL - Via XX Settembre, 31/4 - Tel. 010.586238
ELABORATION PROCESSES SRL - Via Brigata Lig., 68/70/72/74 - Tel. 010.565704
SISTEX SRL - Via SS. Giacomo e Filippo, 13R - Tel. 010.873444
La Spezia - DIFF.EL. CESA SRL - Viale S. Bartolomeo, 139 - Tel. 0187.505223
Sanremo - DIFF.EL. RCS SRL - Via Helsinore, 8 - Tel. 0184.72435
Savona - SISTEX INFORMATICA SRL - Via Montenotte, 100/102 - Tel. 019.801638

LOMBARDIA
Albino - NUOVA INFORMATICA SAS - Via Provinciale, 86 Comenduno - Tel. 035.751784
Assago - TRANSDATA SRL - Mi Fiori Pal. E3 Str. 1 - Tel. 02.8242460
Bergamo - SELTERING SPA - Via Verdi, 31 - Tel. 035.248256
Brescia
FIN-ECO SERVICE SRL - Via Pastrengo, 5 - Tel. 030.59055
MICROSELT SRL - Via Cipro, 33 - Tel. 030.224246
SELTERING SPA - Via Cipro, 33 - Tel. 030.220391
Bresso - C.I.S.I. SAS - V. Vittorio Veneto, 111 - Tel. 02.6105798
Como - BRUNO SRL - Via Rubini, 5 - Tel. 031.260538
Cremona - FIN-ECO SERVICE SRL - P.zza Marconi, 3
Lecco - ZECCA UFFICIO SPA - Viale Dante, 14 - Tel. 0341.373291
Lodi - ZUCCHETTI SPA - C.so Mazzini, 39 - Tel. 0371.54827
Mantova
ANTEK COMPUTER SAS - Via Cavour, 69/71 - Tel. 0376.329333
REPLICA COMPUTER SRL - Via G. Arrivabene, 22 - Tel. 0376.368821
Milano
AG INFORMATICA SRL - Via G. Silva, 49 - Tel. 02.4983416
AMPLIFON SPA-AMPLISISTEM - Via Ripamonti, 129 - Tel. 02.53591
AMUFFICIO SAS - Via Desenzano, 7 - Tel. 02.4080275
B.O.M. SAS DI L. LEPRI & C. - V.le Tunisia, 50 - Tel. 02.6598076
COMPUTER SHARING NORD SRL - Piazza S. Maria Beltrade, 1 - Tel. 02.860586
C.S.A. COMM. SRL - Via Farini, 82 - Tel. 02.6888433
CTC GROUP SRL - Via Dante, 14 - Tel. 02.661450
DATAMONT I.S. SPA - Via Valassina, 22 - Tel. 02.63331
DATA OPTIMATION SRL - Via Masaccio, 12 - Tel. 02.4987876
ECS ITALIA SRL - C.so Monforte, 15 - Tel. 02.780213
EDELEKTRON SRL - C.so Sempione, 39 - Tel. 02.3493603
ELEDRA 3S SPA - Viale Elvezia, 18 - Tel. 02.349751
GENERAL ELECTRIC INFORMATION SERVICES SPA - V.le Regina Giovanna, 29 - Tel. 02.2870181
HOMIC PERSONAL COMPUTER SRL - Piazza De Angeli, 3 - Tel. 02.4988201
HUGNOT LUIGI LUCIANO - Via De Togni, 10 - Tel. 02.873190
IL NUOVO UFFICIO SISTEMI SNC - Via Priv. del Don, 2 - Tel. 02.8350780
ISTITUTO SUPERIORE DI INFORMATICA SRL - Via Montepulciano, 11 - Tel. 02.6701779
ITALSIEL SPA - Via Porlezza, 12 - Tel. 02.3452270
MICROTECH SRL - Via F.lli Bronzetti, 20 - Tel. 02.733609
O.E.M. SRL - C.so Sempione, 8 - Tel. 02.3492136
SIRIO SHOP SRL - Viale Certosa, 148 - Tel. 02.3010051
SOFTEC COMPUTER SRL - Viale Maino, 10 - Tel. 02.7491196
S.D.I. STUDIO DI INFORMATICA SPA - Via G. Winckelmann, 1 - Tel. 02.4227361
Monza - EDICONSULT SRL - Via Rosmini, 3 - Tel. 039.389850
Pavia - I.T.C. INFORMATICA SRL - Strada Nuova, 86 - Tel. 0382.303201
Rozzano - COMPUTER ASSOCIATES SRL - Palazzo Q/8 - Mi Fiori Tel. 02.8242151
Saronno - DATA BASE SISTEMI SRL - Via Don Griffanti, 2 - Tel. 02.9622896
Sondrio - G.P.D. DOMENIGHINI SRL - V.le N. Sauro, 28 - Tel. 0342.218561
Trezzano sul Naviglio - LA CENTRALE SERVIZI SPA - Via B. Cellini, 1 - Tel. 02.445741
Varese
ELMEC SPA - Via Sebenico, 12 - Tel. 0332.264135
I.R.P.E. INF. ELETTR. SRL - Via Morazzone, 8 - Tel. 0332.238533
VEGA SPA - Via Silvestro Sanvito, 103 - Tel. 0332.229374
Vigevano - LOGICA INFORMATICA SRL - Via Montegrappa, 32 - Tel. 0381.81888
Vimercate - DATA PROGRES SRL - Via V. Emanuele, 44/A Tel. 039.667423
Vimodrone - OMEGA DATA SRL - Strada Padana Sup., 317 - Tel. 02.2504121

MARCHE
Civitanova M. - S.E.I. SRL - Via G. D'Annunzio, 198 - Tel. 0733.773262
Fossombrone - SIPCOA COMPUTER SRL - Via Agostini, 3 - Tel. 0721.75340
Jesi - SYSTEM HOUSE A.P.R.A. SRL - V.le Cavallotti, 9 - Tel. 0731.58743
Moie - S.E.D.A. SPA - P.zza S. Maria - Tel. 0731.70345
Pesaro - COMPUTER & OFFICE SRL - Via Mazzini, 73 - Tel. 0721.64170

PIEMONTE
Alba - SISTEMI SRL - Via D. Galimberti, 3/E - Tel. 0173.49871
Alessandria - INFORMATICA SERVICE SRL - Via Isonzo, 63 - Tel. 0131.445817
Asti - HASTA DATI SNC - Via Silvio Morando, 6/A - Tel. 0141.216356
Biella
TEOREMA SRL - Via Losana, 9 - Tel. 015.24915
V.I.P. COMPUTERS SRL - Via Repubblica, 39 - Tel. 015 27106
Borgosesia - I.D.S. INF DATA SYST SRL - Viale Varallo, 157 - Tel. 0163.25327
Cuneo - SISTEMI SRL - Via Giolitti, 26 - Tel. 0171.55475
Genola - EUROSISTEMI SPA - Bivio S.S. 20/28 - Tel. 0172.68176
Novara - ASA SRL - Corso Italia, 25 - Tel. 0321.28250
Torino
ABA ELETTRONICA - Via Fossati, 5/C - Tel. 011.332065
DIVERSIFICATE VENCO SRL - C.so Matteotti, 32A - Tel. 011.545525
PROGRAMMA SPA - C.so Svizzera, 185 - Tel. 011.746421
SISTEMI SPA - C.so Peschiera, 249 - Tel. 011.3358676
SOFTEC SRL - C.so San Maurizio, 79 - Tel. 011.8396444
Verbania (Intra) - S 80 SCRL - Via Roma, 7 - Tel. 0323.41083
Vercelli - ANALOG SNC - Via Dionisotti, 18 - Tel. 0161.61105

PUGLIE
Bari
COMPUTER SHARING SUD SPA - Via Trento, 3 - Tel. 080.339177
H.S. SYSTEMS SRL - Via Castromediano, 131 - Tel. 080.331654
PASED SRL - Via Calefati, 134/136 - Tel. 080.481488
PUBLISISTEMI SRL - Via Castromediano, 131 - Tel. 080.583943
SIRCOM SRL - Via della Repubblica, 67/69 - Tel. 080.364674
Foggia - MASELLI PER L'UFFICIO - Via L. Zuppetta, 355A - Tel. 0881.78014
Lecce - I.P.E.S. SPA - Via Oberdan, 29 - Tel. 0832.33904
Maglie - S.V.I.C. SRL - Via V. Emanuele, 121 - Tel. 0836.21604
Taranto - ALFA di SPORTELLI L. - Via Medaglie d'Oro, 39 - Tel. 099.335558

SARDEGNA
Cagliari
C.D.S. SAS - Via Sonnino, 108 - Tel. 070.650756
DATA SISTEMI SRL - Via Lo Frasso, 6/8 - Tel. 070.662541
Olbia - C.P.S. SRL - Via Galvani, 4 - Tel. 0789.51194

SICILIA
Catania
ASIA COMPUTER SRL - Via S. Eupilio, 13 - Tel. 095.326944
COMPUTER SYSTEMS SRL - Via Ruggero di Lauria, 87 - Tel. 095.493777
Messina - SO.F.IN. SPA - Via Don Blasco, 75 - Tel. 090.2923987
Palermo
SER.COM. ITALIA SRL - Via Sciuti, 180 - Tel. 091.261041
SI.PR.EL. SRL - Via Serradifalco, 145 - Tel. 091.577344
TESI SRL - Via E. Notarbartolo, 23 - Tel. 091.260549
Trapani - TESI SRL - Via Palmerio Abate, 2 - Tel. 0923.20026

TOSCANA
Arezzo - FINITAL FIN. IT. SPA - Via Benedetto Varchi, 59 - Tel. 0575.353141
Empoli - SESA DISTRIBUZIONE SRL - Via XI Febbraio, 24/B - Tel. 0571.72148
Firenze
DATA COOP SCRL - Via di Novoli, 23/H - Tel. 055.4379868
DISTAL SRL - Via Pacini, 46 - Tel. 055.350669
SESA DISTRIBUZIONE SRL - Lungarno Ferrucci, 19R - Tel. 055.6811652
Pistoia - DARIO NANNINI - Corso Gramsci, 84
Prato - C.C.S. SAS - Viale Repubblica, 298 - Tel. 0574.580222
Siena - SILOG SRL - Via Sicilia, 5 - Belverde Monteriggioni - Tel. 0577.54085
Viareggio - DELPHI SRL - Via Aurelia Sud, 39 - Tel. 0584.395161

TRE VENEZIE
Bassano D/Grappa - C.P.E. - Piazzetta Poste, 9 - Tel. 0424.20395
Belluno
DE PRÀ SRL - Via I. Caffi, 18 - Tel. 0437.23243
SCP. COMP. SYST. SRL - Via Feltre, 32 - Tel. 0437.20826
Bolzano - BOPAM SAS - Via C. Battisti, 32 - Tel. 0471.30113
Brunico - DATOR SRL - Via Campo Tures, 8 - Tel. 0474.84815
Castelfranco Ven. - EDS SRL - Via S. Pio X, 154 - Tel. 0423.490178
Mestre - BOFFELLI F.LLI G & E SNC - Via Bissolati, 19/21/23 - Tel. 041.951247
Padova
CERVED ENGINEERING SPA - C.so Stati Uniti, 14 - Tel. 049.760733
S.I.C. ITALIA SRL - Via S. Pietro, 82 - Tel. 049.34394
SYSTEM ROS SAS - P.zza De Gasperi, 14 - Tel. 049.38412
SO.GE.DA. SPA - Via Marsala, 29 - Tel. 049.655385
S. Donà di Piave - COMPUTIME SRL - Piazza Rizzo, 63 - Tel. 0421.2548
Schio - LINEA 4 C.S.N.C. - Via Riva del Cristo, 4/6/8 - Tel.
Trento
ATR DESIGN COLL. SRL - Via Torre Verde, 25 - Tel. 0451 26872
SEDA SAS - Via Sighele, 7/1 - Tel. 0461 984564
SiGe SNC - COMPUTER SHOP - Via Prato, 22 - Tel. 0461.25154
Treviso
DATUM SRL - Via S. Bona Nuova, 3 - Tel. 0422.22560
INFORMATICA TRE SRL - Viale della Repubblica, 19 - Tel. 0422.65993
Trieste
DITTA MURRI - Via A. Diaz, 24/A - Tel. 040.733253
SELTED SRL - Via Fabio Filzi, 23 - Tel. 040.61381
Udine
D.E.U. SRL - Via Di Prampero, 3/7 - Tel. 0432.204402
D.E.U. SRL - Via Tavagnacco, 89 - Tel. 0432.482086
Verona
PRAGMA SOFTWARE SRL - Via Carmelitani Scalzi, 20 - Tel. 045.596400
SEVER DI G. SERENI - Via Locatelli, 10 - Tel. 045.31331
Vicenza - ALFA DATA SRL - Via Milano, 110 - Tel. 0444.31865

UMBRIA
Perugia
PRISMA INFORMATICA SRL - Via Campo di Marte, 4 N - Tel. 075.71973
PUCCIUFFICIO SNC - Via XX Settembre, 148/C - Tel. 075 72992
Terni - DPS SRL - Via Pacinotti, 6 - Tel. 0744.58247

VALLE DI AOSTA
Aosta - INFORMATIQUE SAS - Centre Commercial l'Amerique, S.S. 26 - Quart d'Aosta - Tel. 0165.765173

IBM Italia
Distribuzione Prodotti S.p.A.

IBM

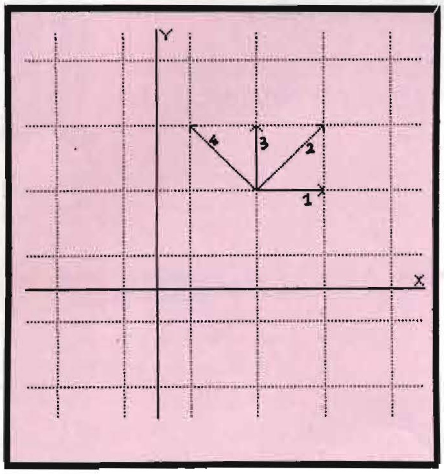


Figura 7. *I diversi modi di passare da un punto del reticolo al punto vicino: essi danno origine alla formula (1.1 - 1.4).*

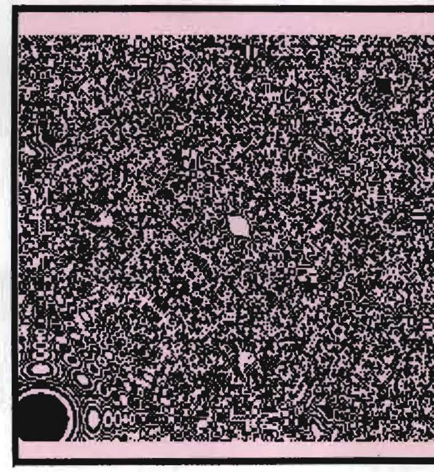


Figura 9. *Per capire come mai compare la zona bianca lungo la diagonale occorrerebbe analizzare anche la derivata dell'intervallo di campionamento al passare da un punto al vicino.*

niamo" questi valori di t in modo che la distanza tra l'uno e l'altro sia minore o uguale alla metà del periodo della funzione, cioè a π, i punti che ricaviamo in questo modo riescono a riprodurre bene tutte le oscillazioni della funzione (la sinusoide a frequenza più bassa che passa per i punti ottenuti è proprio quella che ci è servita per ricavarli) (figura 6a). Ma se la distanza tra un valore di t e il successivo è maggiore di π, i punti ricavati non riescono più a seguire le oscillazioni che la funzione compie (si può trovare una sinusoide con frequenza più bassa della nostra che passa per tutti i punti ottenuti) (figura 6b). In tale caso i punti che abbiamo ricavato riproducono invece molto meglio le oscillazioni di un'altra sinusoide, con periodo più lungo di quella dalla quale li avevamo ricavati.

È interessante vedere che cosa succede in due casi particolari: quando l'intervallo di campionamento è poco minore (o maggiore) di 2π, e quando è uguale a $2\pi/3$.

Nel primo caso (figura 6d) il periodo della sinusoide a frequenza più bassa passante per i punti dati diventa molto grande rispetto a quello della sinusoide originale, e si può osservare come vi sia un gran numero di punti campionati consecutivi con lo stesso segno.

Nel secondo caso (figura 6c) si ha una distribuzione regolare di due punti campionati vicini dello stesso segno, seguiti da uno di segno opposto, e così via.

Abbiamo allora visto come, quando l'intervallo di campionamento è troppo grande, non è possibile riprodurre bene l'andamento di una funzione che oscilla in segno. Come adesso vedremo, questa situazione si manifesta proprio nelle nostre figure.

Nel nostro caso, dato che stiamo rappresentando la funzione $\sin(r^b)$, al posto di t usiamo r^b, e l'intervallo di campionamento, che prima era la distanza tra un valore di t ed il successivo, adesso diventa la differenza dei valori di r^b calcolati in punti adiacenti del reticolo. Come si vedrà nell'analisi seguente, questa differenza non è costante, ma varia al variare di x e di y.

Per semplicità analizziamo per ora solo il caso b = 2, e calcoliamo il valore del nostro intervallo di campionamento. Questo è dato dalla variazione di r^2 nel passare da un puntino i, j ai puntini vicini. Ci sono 4 modi diversi di spostarsi da un puntino al vicino (figura 7): esaminiamo ognuno di questi modi

1) $r^2(i+1, j) - r^2(i, j) = r^2(x+\Delta x, y) - r^2(x, y) = x^2 + 2x\Delta x + (\Delta x)^2 + y^2 - x^2 - y^2 = 2x\Delta x + 0((\Delta x)^2)$ (1.1)

dove con $0((\Delta x)^2)$ indico termini del secondo ordine in Δx, perciò trascurabili

2) $r^2(i+1, j+1) - r^2(i, j) = 2x\Delta x + 2y\Delta x$ (1.2)

3) $r^2(i, j+1) - r^2(i, j) = 2y\Delta x$ (1.3)

4) $r^2(i-1, j+1) - r^2(i, j) = 2y\Delta x - 2x\Delta x$ (1.4)

Da questi risultati si vede come l'intervallo di campionamento dipenda dai valori di x e y (oltre che da Δ x, che però è costante). Procedendo ad esempio lungo la retta y = 0, e allontanandosi dall'origine, x cresce, e con esso crescono anche gli intervalli di campionamento nelle direzioni 1, 2 e 4 fino ad oltrepassare il valore di π; oltre questo punto l'andamento della funzione non è più ben rappresentato dai punti disegnati sul video. Questo del resto lo avevamo già intuito notando come le larghezze degli anelli circolari, che rappresentano gli intervalli di r per i quali la funzione conserva lo stesso segno, diminuiscano a mano a mano che ci si allontana dal centro della figura. Quando gli anelli hanno spessore uguale ad un puntino, siamo nella zona in cui l'intervallo di campionamento nella direzione del raggio è vicino a π (la funzione cambia segno ad ogni puntino). Oltrepassato il raggio per il quale l'intervallo è uguale a π, la struttura ad anelli via via più sottili non può più rappresentare bene la funzione.



FESTA NAZIONALE DE L'UNITÀ - ROMA 30 AGOSTO - 15 SETTEMBRE 1984
computer games 84
IL PRIMO FESTIVAL DEGLI AUTORI DI SOFTWARE PER IL GIOCO
"PERCHÈ I COMPUTERS NON SI PRENDANO GIOCO DELLA VITA"
10 BUONE REGOLE PER PARTECIPARE A QUESTO CONCORSO:
1) Realizzate programmi originali. Per favore evitate le copie dei videogames da bar e stanche storie di invasioni di qualsiasi specie terrestre o spaziale.
2) Scriveteli nel linguaggio che meglio conoscete. Se scrivete in Basic fatelo con stile.
3) Qualunque sia il tipo di gioco - di strategia, di avventura, animato, educativo - cercate di realizzare il maggior numero di schermi. La varietà li renderà più interessanti.
4) Fateli su home o personal computer, su disco o cassetta.
5) Realizzate una buona pagina, di presentazione, con un titolo e in seguito le istruzioni.
6) Dedicate molto tempo al "DEBUG", cioè alla eliminazione degli errori e al perfezionamento.
7) Fateli per il computer in versione di memoria base, senza espansione. Se ben utilizzati, vi consigliamo l'uso di accessori per l'input, come il joystick, la penna ottica, la tavoletta grafica, ecc.
8) Fate il maggior sforzo possibile per dare un tocco di stile, non solo nella realizzazione della dinamica del gioco ma anche nell'accuratezza degli scenari, nel disegno dei caratteri, nella velocità di animazione, negli effetti sonori.
9) Mandateci il programma. Non avete niente da perdere. La nostra giuria deciderà se il vostro programma entrerà nella rosa finale, che verrà presentata alla Festa Nazionale de l'Unità a Roma, dal 30 Agosto al 15 Settembre 1984. In quell'occasione, al voto della giuria si aggiungerà quello del pubblico: sarà così designato il programma vincitore di questa PRIMA EDIZIONE DEL COMPUTER GAMES FESTIVAL. Se il vostro programma non sarà selezionato per le finali, vi sarà restituito con una lista di osservazioni e consigli. Ai vincitori, oltre alla pubblicazione dei loro programmi, verranno dati favolosi premi consistenti in monitors televisivi a colori e videoregistratori.
10) Ultima regola: siete chiamati a sfidare il computer con la vostra creatività.
LABORATORI DI SVILUPPO
a cura dell'ARCI
Durante i mesi di Maggio, Giugno e Luglio saranno attivati laboratori come supporto per l'elaborazione di computer games, corsi di basic e computer-graphic, Bologna, Torino, Roma, Milano, Firenze, Napoli, Bari, Palermo, Cagliari.
PER RICEVERE LA SCHEDA DI PARTECIPAZIONE E LE INFORMAZIONI RELATIVE AL CONCORSO SCRIVERE O TELEFONARE A:
MAGIC BUS CAS. POST. 1144
40100 BOLOGNA
TEL. 051/228937-223708.
GRUPPO EDITORIALE JACKSON
Italia: Via Rosellini, 12 - 20124 Milano
Stati Uniti: GEJI Publishing Group, Inc. 1143 Quince Avenue Sunnyvale, CA 94087
Gran Bretagna: GEJ Publishing Ltd Oxford Street 187 London WIRIAJ

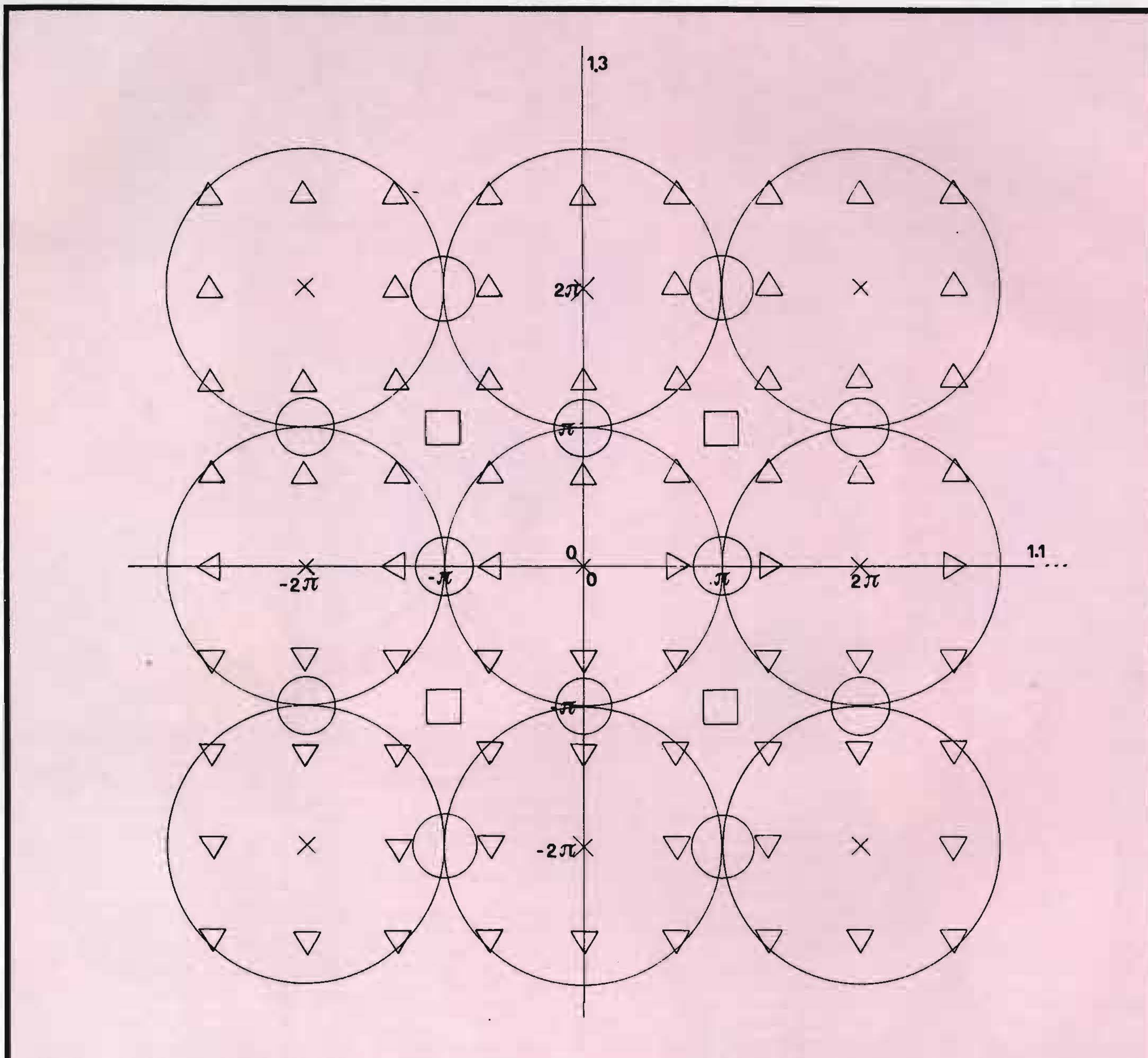


Figura 8. *Lo schema di figura illustra la struttura delle zone di simmetria più evidenti visibili, ad esempio, in figura 3c. I punti marcati con la croce sono quelli nei quali i 4 valori (1.1 - 1.4) degli intervalli di campionamento sono multipli interi di 2 π, e i cerchi che li contengono rappresentano l'estensione delle zone a simmetria circolare centrate in questi punti.*

In tabella 1 sono riportati, per b = 2 e per diversi valori di XM, i valori di x per i quali succede quanto detto; si può facilmente verificare come le figure siano in pieno accordo con questi risultati. Si noti, ad esempio, come nel caso della figura 3a, nella quale la simmetria circolare si conserva indisturbata, il punto in questione giaccia molto fuori dello schermo, mentre nella figura 3b si comincino a notare delle irregolarità dovute alla vicinanza del punto in questione (anche in questo caso fuori dello schermo).

Sorge però adesso una domanda imbarazzante; abbiamo visto cosa succede sull'asse x, e come, oltrepassata una certa distanza dall'origine, la distanza tra i punti del reticolo renda impossibile riprodurre bene l'andamento del segno della funzione. Questo non spiega ancora come mai, dato che la funzione dipende solo dalla distanza dall'origine, il suo comportamento non è uguale in tutte le direzioni, e cioè

perché quello che avviene sull'asse x non si riproduca esattamente su un altro qualsiasi raggio.
La causa di questa rottura di simmetria sta nel fatto che, anche se la funzione che viene rappresentata ha una simmetria circolare, il reticolo che si usa per definire i punti del piano xy nei quali il valore della funzione verrà calcolato per rappresentarla non possiede questa simmetria. In altre parole: è vero che il valore della funzione dipende solo da r, ma nel reticolo che usiamo, pochi punti hanno lo stesso valore di r (figura 2c). Se invece di un reticolo come quello che usiamo (al quale siamo del resto vincolati perché il reticolo dei puntini che possiamo visualizzare sul video ha la stessa forma) potessimo usare il reticolo di figura 2d, potremmo continuare ad osservare una simmetria circolare nella figura.
In pratica, usando un reticolo quadrato invece che quello circolare, si commette un errore dello stesso tipo di quello che si commetterebbe se nelle figure 6, invece di usare per il campionamento i valori di t dai quali abbiamo ricavato i punti marcati con le croci, avessimo usato degli altri valori, ognuno dei quali compreso nell'intervallo tra due dei valori originali, ma non ad una distanza fissa tra essi.

Ma allora, visto che il reticolo usato è inadeguato a rappresentare una funzione a simmetria circolare, come mai nella parte centrale delle figure 3 questa simmetria continua ad essere osservata, e come mai nella figura 3a non si osserva alcuna rottura di simmetria?
La risposta è abbastanza semplice: nella parte centrale della figura la situazione è analoga a quella di figura 6a, e il fatto di usare un puntino del reticolo quadrato invece che uno del reticolo circolare non modifica sensibilmente la rappresentazione della funzione; invece, dove la distanza tra due punti vicini del reticolo è dello stesso ordine di grandezza del periodo della funzione (figura 6d), l'errore che si commette usando il reticolo quadrato può alterare completamente la rappresentazione.

La simmetria residua rispetto agli assi ed alle diagonali

Questa simmetria residua, osservabile in ognuna delle figure 3, deriva dal fatto che, poiché il reticolo di punti sul quale calcoliamo il valore della nostra funzione è centrato nell'origine delle coordinate, per ogni punto del reticolo (x, y) associato ad una coppia di valori (i, j) esistono effettivamente sul reticolo i punti ad esso simmetrici rispetto agli assi e alle diagonali (figura 4a). I valori di x e di y in questi altri punti si possono ottenere o scambiando i segni di x, di y o di entrambi, oppure scambiando x con y (e poi scambiando i segni come nel caso precedente).

```
2x△x = π                (△x=2XM/215)

        ---------------------

da cui  I      x= π/2△x       I

        ---------------------


      XM        I         x
    ------------I------------
       6        I       28.14
      10        I       16.89
      20        I        8.44
      30        I        5.63
      50        I        3.38
      70        I        2.41
     100        I        1.69
```

Tabella 1. *Distanza dall'origine oltre la quale non è più riconoscibile la simmetria circolare (figure 3a-3g).*

Poiché la funzione dipende solo da r, e cioè da $x^2 + y^2$, è chiaro che nei punti suddetti il valore sarà lo stesso. Per eliminare questa simmetria, basta spostare l'origine delle coordinate in modo da distruggere questa corrispondenza (figura 4b). Ciò si ottiene molto facilmente ad esempio modificando la linea:

1023 X = X + 0.05 : Y = Y + 0.09

Facendo allora girare il programma con b = 2 e XM = YM = 30, si ottiene la figura 5; si può notare come la simmetria quadrata sia scomparsa: ad esempio, le zone centrali dei 4 cerchi sopra, sotto e di fianco a quello contenente l'origine sono diverse tra loro, mentre prima erano uguali (figura 3d).

Perché compaiono altre zone a simmetria circolare

Dalle formule (1.1) - (1.4) si nota come esistano dei punti nei quali ognuno dei 4 intervalli di campiona-

Indagine su alcune strane figure

mento vale 0, o 2 π, o un multiplo intero di 2 π; questi punti sono quelli per i quali sia x che y si possono scrivere nella forma

$$\frac{2 n \pi}{2 \Delta x},$$

con n intero.
In questi punti si ha una situazione analoga a quella dell'origine; inoltre, poiché la variazione dell'intervallo di campionamento passando da un puntino al vicino è costante in tutto lo spazio (lo si vede differenziando le 1.1 - 1.4), lo spazio necessario perché l'intervallo di campionamento passi da 0 a π è uguale a quello per passare da 2 π a 3 π, e così le dimensioni degli anelli che si vengono a formare intorno a questi punti "privilegiati" sono uguali in tutta la figura.
Il colore e la dimensione degli anelli centrali possono essere diversi in ognuno di questi punti; ciò dipende dallo sfasamento che la funzione sin (r^2) ha in questi punti rispetto all'origine, sfasamento che è facilmente calcolabile:

$$\varphi = r^2 \bmod 2\pi = \left[\frac{4\pi^2}{4(\Delta x)^2}(n^2_x + n^2_y)\right] \bmod 2\pi$$

In figura 8 è riportato uno schema che rappresenta la struttura delle nostre figure in funzione degli intervalli di campionamento 1 e 3 (gli intervalli 2 e 4 si ottengono rispettivamente sommando e sottraendo 1 e 3). In questo schema i punti marcati con "x" rappresentano i punti dei quali si è detto sopra, e i cerchi che li contengono sono le zone a simmetria circolare ben definita intorno a questi punti.

Sulle figure notiamo, molto evidenti anche se meno delle prime, altre zone con una certa simmetria circolare. Queste zone si trovano nei punti dove 3 degli intervalli di campionamento valgono +/− 2 π/3, + 2 n π, ed il restante 2 n π. Sul nostro schema contrassegnamo questi punti con un triangolo.
Queste zone sono meno evidenti delle precedenti sia perché quando l'intervallo di campionamento vale 2 π/3 non si ha un gran numero di punti positivi seguiti da un gran numero di punti negativi, ma la distribuzione a due punti di un colore seguiti da un punto del colore opposto già illustrata in figura 6c, sia perché il fatto che uno degli intervalli valga 2 n π fa si che in una direzione il valore della funzione mantenga lo stesso segno per un buon numero di puntini, disturbando la simmetria circolare.
I punti contrassegnati con un cerchietto sul nostro schema sono quelli nei quali 3 degli intervalli valgono 2 n π + π, ed il quarto vale 0 (o 2 n π). Così, mentre l'intervallo uguale a 0 è responsabile delle linee continue che attraversano la figura, i tre intervalli uguali a π fanno si che, nella direzione perpendicolare a queste linee, puntini vicini abbiano segni opposti e, perciò, colori diversi.

Infine, nei punti contrassegnati da un quadratino, gli intervalli 1 e 3 valgono π, e così ogni puntino è di colore opposto a quello immediatamente a fianco, o sopra, o sotto, mentre gli intervalli 2 e 4 valgono 2 n π, dando origine alle linee diagonali formate da puntini dello stesso colore.

Cosa succede quando b è diverso da 2

Osservando le figure 3h-3p, ottenute con valori di b diversi da 2, si possono notare 3 cose:
1) le varie zone che si individuano nelle figure non hanno più una simmetria circolare, ad eccezione della zona centrale;
2) le dimensioni delle zone variano all'interno delle singole figure; in particolare, se b > 2 le zone diventano più piccole allontanandosi dal centro, mentre se b < 2 succede il contrario;
3) anche la densità delle zone varia all'interno delle figure, e si perde la struttura a reticolo ripetitivo che caratterizza il caso b = 2. Quando b > 2 le zone sulle diagonali tendono ad avvicinarsi al centro, mentre quando b < 2 si allontanano da esso.

Tutti e tre questi fatti sono facilmente spiegabili osservando l'andamento degli intervalli di campionamento e delle loro variazioni in funzione di x e y.
Generalizzando le espressioni 1.1 - 1.4 si ha:

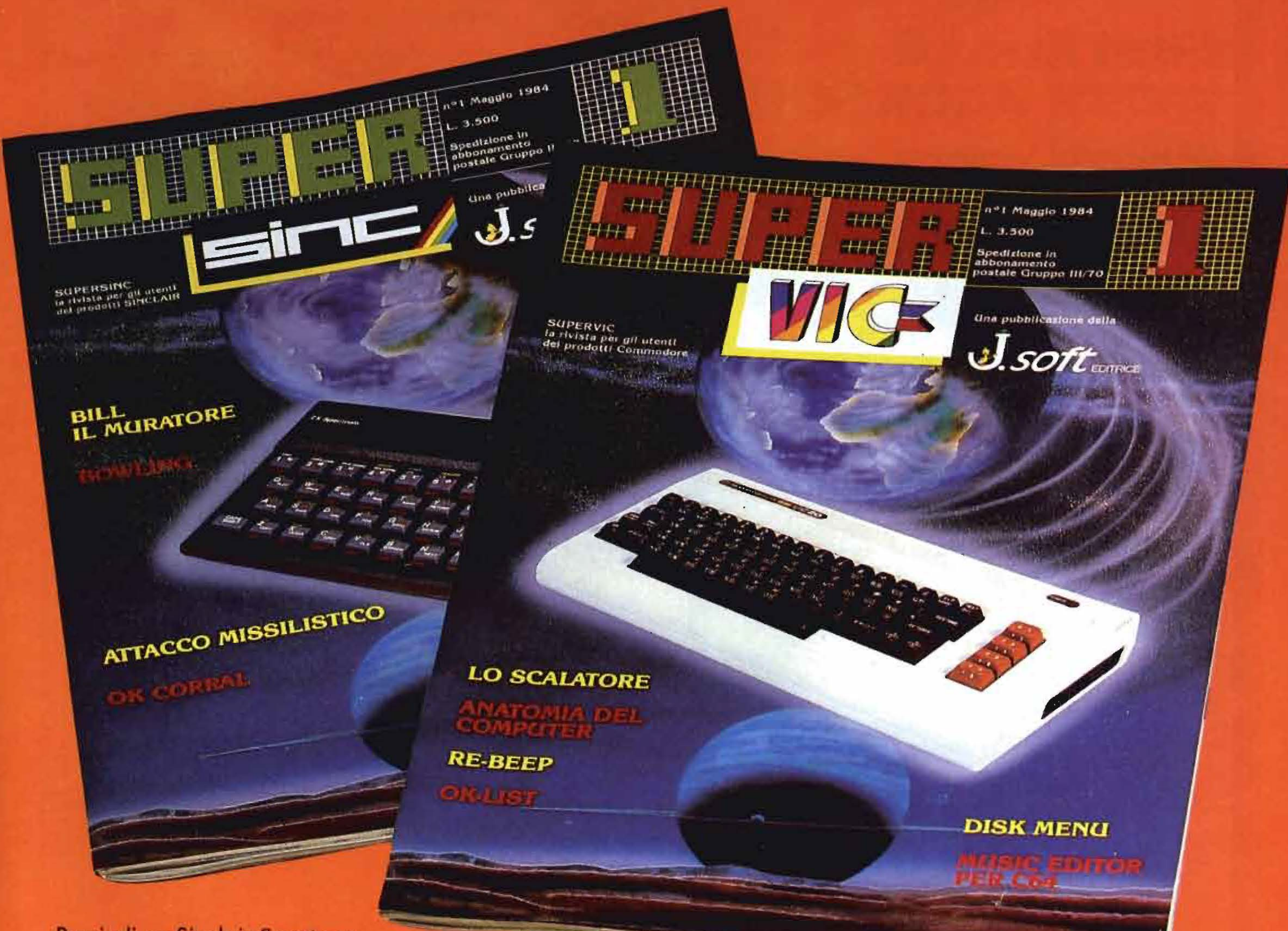
$$r^n(x + \Delta x, y) - r^n(x, y) = [r^2(x + \Delta x, y)]^{n/2} - [r^2(x, y)]^{n/2} =$$
$$= (x^2 + 2 x \Delta x + y^2)^{n/2} - (x^2 + y^2)^{n/2} \equiv \Delta R_x$$

mentre, per le variazioni dell'intervallo di campionamento:

```
1000  READ B,XM
1002  YM=XM
1010  MODE 6:COLORG 15 3 5 0
1015  FILL 0,0 XMAX,YMAX 15
1020  FOR I=0.0 TO 215.0:FOR J=0.0 TO 215.0
1022  X=I*2.0*XM/215.0-XM:Y=J*2.0*YM/215.0-YM
1024  R=SQR(X*X+Y*Y)
1025  Z=SIN(R^B)
1030  COLOR%=0:IF Z<0 THEN COLOR%=15
1040  DOT I,J COLOR%
1050  NEXT J:NEXT I
1060  END
```

Listato 1. *Programma per il tracciamento dei grafici scritto in BASIC per il DAI, ma facilmente traducibile per altri calcolatori.*

SONO IN EDICOLA



Possiedi un Sinclair Spectrum o uno ZX81 e vuoi saperne di più? Vuoi disporre di nuovi programmi, giochi e idee per utilizzare al meglio il tuo piccolo gioiello? Corri in edicola e acquista SUPERSINC per dare una nuova carica al tuo home computer. SUPERSINC, in edicola tutti i mesi, è la rivista nuova e indispensabile per chi possiede un computer Sinclair.

SUPERVIC è il "carburante" per il tuo VIC 20 e il tuo C 64. SUPERVIC è la nuova rivista pensata e realizzata per utilizzare ancora meglio il tuo personal computer. SUPERVIC ti propone, ogni mese giochi, utility, software tools, notizie, prezzi. E tante idee nuove! Il pieno di software si fà in edicola con SUPERVIC.

J.soft

Un marchio di TechnoClub s.r.l.

Via Rosellini 12, 20124 Milano tel. 6880951 TLX 333436

...Idee nuove per il tuo personal.

Indagine su alcune strane figure

$$\frac{j}{jx} \Delta R_x = (2x + 2\Delta x)(x^2 + 2x\Delta x + y^2)^{n/2 -1} - 2x(x^2 + y^2)^{n/2 -1}$$

$$\frac{j}{jy} \Delta R_x = (2y)(x^2 + 2x\Delta x + y^2)^{n/2 -1} - 2y(x^2 + y^2)^{n/2 -1}$$

Come si vede, la variazione dell'intervallo di campionamento al passare da un punto al vicino dipende da x e da y; se $b > 2$ essa cresce al crescere della distanza dall'origine, e ciò spiega la diminuzione delle dimensioni delle zone nelle figure 3h-3j, 3n e 3p; al contrario, se $b < 2$, la variazione decresce con l'allontanarsi dall'origine, e così le zone diventano più grandi.

La perdita della simmetria circolare delle singole zone dipende invece dal fatto che la variazione dell'intervallo di campionamento calcolata in un dato punto dipende dalla direzione lungo la quale questa variazione è calcolata.

Infine, la deformazione del reticolo è causata dal diverso andamento dell'intervallo di campionamento; in particolare, mentre nel caso $b = 2$ valeva l'uguaglianza:

$$x^b + y^b = r^b,$$

in generale si ha che il primo termine è maggiore del secondo se $b < 2$, mentre è minore se $b > 2$. Questo spiega l'avvicinamento al centro delle zone lungo le diagonali, rispetto alle zone lungo gli assi x e y, nel caso di $b > 2$, ed il loro allontanamento nel caso contrario.

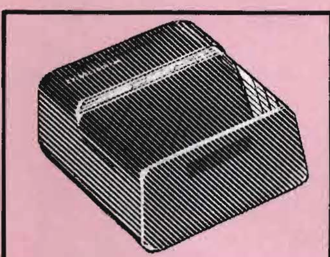
Conclusioni

Spesso problemi apparentemente semplici e privi di difficoltà, come ad esempio tracciare il grafico di una funzione, nascondono complicazioni alle quali occorre dedicare un po' di attenzione. L'analisi svolta in questo articolo è stata volutamente poco rigorosa e molto intuitiva, perché le complicazioni matematiche non facessero perdere di vista il punto centrale del problema, ossia la spiegazione dell'origine delle figure 3a-p, ormai non più tanto strane. In particolare, si sono trascurati gli effetti degli errori di arrotondamento introdotti dal calcolatore nel rappresentare i numeri reali (questi errori non disturbano nel caso $b = 2$, ma non si potrebbero trascurare al crescere di b e di XM), e non si è cercato di capire cosa succede quando anche le variazioni degli intervalli di campionamento subiscono notevoli cambiamenti da un puntino al successivo (figura 9). A volte, come nel nostro caso, alcuni di questi effetti hanno una importanza secondaria e si possono trascurare; altre volte ciò non è possibile, e per risolvere un problema non si può evitare di analizzarne ogni minimo particolare. ■

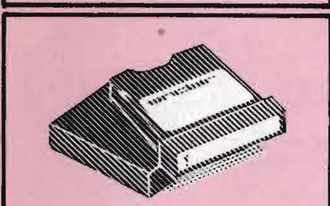
ZX Microdrive



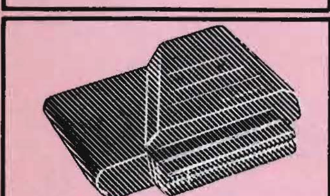
fai crescere il tuo sinclair - Spectrum con le sue eccezionali periferiche!



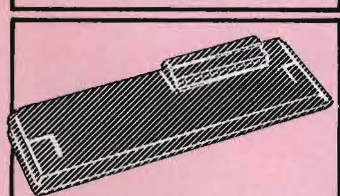
ZX MICRODRIVE
Amplia le possibilità dello ZX Spectrum in quei settori come la didattica e le piccole applicazioni gestionali, dove è necessaria una ricerca veloce delle informazioni.
Ogni cartuccia può contenere:
85 kbyte / 95 kbyte



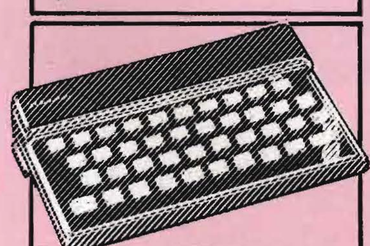
CARTRIDGE
Per ZX Microdrive.
Capacità:
85 kbyte / 95 kbyte
Confezione da 2 pezzi.



ZX INTERFACE 2
Permette di utilizzare le nuovissime ZX ROM, cartucce software e il collegamento per 2 joystick.



ZX INTERFACE 1
Indispensabile per il collegamento dello ZX Microdrive.
Inoltre permette il collegamento fra lo ZX Spectrum e una ampia gamma di periferiche e di altri Sinclair in rete locale.



COMPUTER ZX SPECTRUM
A colori, collegabile ad un televisore a colori o in b/n e ad un normale registratore a cassetta.
32x24 caratteri.
RAM di base: 16 k - 48 k
256x192 punti.
8 colori - 2 luminosità.

sinclair

a casa vostra subito !!

Descrizione	Q.tà	Prezzo unitario	Prezzo Totale
COMPUTER ZX SPECTRUM 16 kbyte		L. 398.000	
COMPUTER ZX SPECTRUM 48 kbyte		L. 499.000	
ZX MICRODRIVE		L. 199.500	
ZX INTERFACE 1		L. 199.500	
ZX INTERFACE 2		L. 95.500	
2 CARTRIDGE		L. 45.000 La coppia	

Desidero ricevere il materiale indicato nella tabella, a mezzo pacco postale contro assegno, al seguente indirizzo:

Nome
Cognome
Via
Città
Data C.A.P.
Desidero ricevere la fattura SI NO
Partita I.V.A. o, per i privati Codice Fiscale

PAGAMENTO:
A) Anticipato, mediante assegno circolare o vaglia postale per l'importo totale dell'ordinazione.
B) Contro assegno, in questo caso, è indispensabile versare l'acconto di Lire 50.000 mediante assegno circolare o vaglia postale. Il saldo sarà regolato contro assegno.
AGGIUNGERE: L. 5.000 per contributo fisso. I prezzi sono comprensivi di I.V.A. 18%

l'unica organizzazione europea per la spedizione di computer e componenti elettronici

EXELCO Via G. Verdi, 23/25
20095 - CUSANO MILANINO - Milano



Calcolo letterale di polinomi

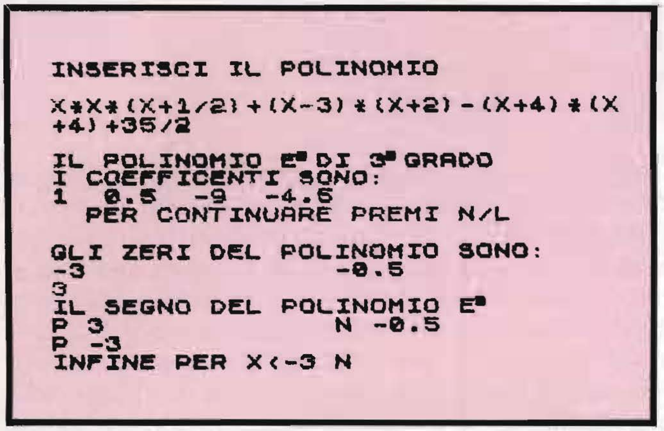
Lo ZX81 amico fedele nei compiti di algebra

di *Carlo Cappelli*

La manipolazione algebrica dei polinomi può sembrare argomento di facile trattazione, poiché tali sembrano le regole che usiamo per riordinare i polinomi a mente; non lo è, invece, se cerchiamo di implementare le stesse regole su un personal computer servendoci del linguaggio BASIC.
Le difficoltà che si incontrano riguardano la complessità degli algoritmi da usare, che comportano tempi di elaborazione proibitivi.
Per rendersi conto di ciò, possiamo analizzare alcune fasi che mentalmente vengono eseguite senza difficoltà, ma che richiedono lunghi tempi di elaborazione per un computer. Prendiamo, ad esempio, il procedimento di somma degli elementi di un polinomio che prevede la somma di tutti i termini di ugual grado (tutti i termini al cubo vengono sommati tra di loro, così pure quelli al quadrato, ecc.).
La macchina non può analizzare tutti i termini del polinomio con uno sguardo d'assieme, ma deve studiarli carattere per carattere, tenendo conto di tutti i modi in cui ogni termine può essere scritto. Ad esempio, X al quadrato può essere scritto X ↑ 2 (↑ indica l'elevamento a potenza) oppure X ★ X che matematicamente hanno lo stesso significato, ma simbolicamente assumono forma diversa. Già per risolvere questi passaggi un normale computer programmato in BASIC impiega tempi lunghissimi, che diventano inaccettabili se deve anche sviluppare prodotti tra polinomi, cosa molto comune nel calcolo letterale. Una soluzione possibile sarebbe di utilizzare il linguaggio macchina, al fine di ottenere una maggiore velocità di esecuzione, ma così facendo si comprometterebbe l'adattabilità degli algoritmi.
Un'alternativa interessante consiste nello sfruttare la caratteristica basilare del computer, intendendolo cioè come struttura di calcolo diversa nella sostanza dalla struttura del pensiero umano che è decisionale, e applicare quindi regole diverse da quelle che noi utilizziamo, usufruendo al massimo dell'enorme velocità con cui vengono eseguiti calcoli anche molto complessi.
Nella pratica del caso in esame è necessario sviluppare algoritmi che conducano al polinomio risultante solo mediante calcoli numerici.

Il programma presentato in questo articolo risponde a queste esigenze; esso svolge rapidamente, in forma esplicita, polinomi in una variabile fino al quinto grado e determina inoltre gli zero (cioè le soluzioni) ed il segno dell'equazione risultante.
Il programma è scritto in BASIC per ZX81 e Spectrum, ma è stato fatto in modo da poter essere adattato facilmente su qualsiasi altro computer.
Il tempo di elaborazione è molto contenuto: per risolvere un polinomio con decine di parentesi occorrono 15 secondi, o anche meno, mentre per trovare gli zero dell'equazione risultante occorrono da 20 a 40 secondi, a seconda del numero di soluzioni da trovare.
Il principio su cui si basa il programma è che inserendo al posto della variabile un valore numerico se ne ottiene un altro, che è specifico di quel polinomio; quindi, in base ai vari risultati ottenuti, inserendo una certa quantità di valori numerici, è possibile determinare l'equazione risultante nella forma esplicita, cioè la più sintetica, senza parentesi e ripetizioni di termini di ugual grado.
Per chi ha qualche nozione di analisi matematica posso dire che ogni polinomio può essere inteso come il secondo membro di una funzione in Yo; dando alla variabile indipendente vari valori si ottengono in corrispondenza altrettanti valori della variabile dipendente. Queste coppie di valori individuano dei punti per i quali passa il grafico della funzione ed in base ad essi si può ricostruire la funzione stessa. Se il polinomio è di I° grado occorrono due punti, se di II° grado tre punti e così via.
Analizziamo ora in dettaglio il programma.
Il polinomio viene inserito in input e deve avere matematicamente senso, cioè deve essere scritto sfruttando i simboli matematici del computer; ciò che comunemente si legge "tre X due", corrisponde a 3 x X ^ 2 e non a 3 X 2. Come variabile si può usare solo la lettera X, altrimenti si devono apportare varie modifiche al programma.
Il polinomio può essere inserito così com'è oppure sotto forma di equazione, divisa in due parti separate da un "="; ci pensa il programma a spostare verso sinistra, cambiando di segno, ciò che sta a destra dell'uguale. Successivamente viene applicato un semplice algoritmo per determinare il grado del polinomio (per chi non lo sapesse il grado di un polinomio non è altro che il valore della potenza più alta. Esempio il polinomio X + 4 x X ^ 2 − X ^ 3 ha grado 3).
In pratica si attribuisce ad X un va-

Calcolo letterale di polinomi



```
INSERISCI IL POLINOMIO
X*X*(X+1/2)+(X-3)*(X+2)-(X+4)*(X
+4)+35/2
IL POLINOMIO E' DI 3° GRADO
I COEFFICENTI SONO:
1  0.5  -9  -4.5
 PER CONTINUARE PREMI N/L
GLI ZERI DEL POLINOMIO SONO:
-3              -0.5
3
IL SEGNO DEL POLINOMIO E'
P 3             N -0.5
P -3
INFINE PER X<-3 N
```

Figura 1. *Quadro video presentato alla fine delle elaborazioni del primo esempio.*



```
INSERISCI IL POLINOMIO
(-X*X+8/3)*(X-5)*(3*X+6)+X*X*(3*
X*X-1)=4*X*X+(X+3/2)*(X-5)
IL POLINOMIO E' DI 3° GRADO
I COEFFICENTI SONO:
9  32  -20.5  -72.5
 PER CONTINUARE PREMI N/L
GLI ZERI DEL POLINOMIO SONO:
-3.5597           -1.5022
1.5064
IL SEGNO DEL POLINOMIO E'
P 1.5064          N -1.5022
P -3.5597
INFINE PER X<-3.5597 N
```

Figura 2. *Quadro video presentato alla fine delle elaborazioni del terzo esempio.*

lore grande, nel nostro caso 1E7, e si divide il risultato per X elevato a un indice G, che viene incrementato ogni volta; quando il risultato della divisione è in valore assoluto molto più piccolo di 1E7 (nel nostro caso quando è minore di 100) allora l'indice G rappresenta il grado del polinomio. Stabilito questo si possono determinare i punti sufficienti a individuare il polinomio risultante, che devono essere in numero di G + 1. Lasciamo ai matematici le considerazioni sui limiti di questo metodo.

Per semplicità inseriamo al posto dei termini in X valori numerici interi a partire dallo zero e ne valutiamo il risultato.

Per rendersi conto di come funziona il programma conviene analizzare il caso pratico di un polinomio di 3° grado:

a x X ^ 3 + b x X ^ 2 + c x X + d (con a, b, c, d coefficienti numerici del polinomio).

Inserendo 0, 1, 2, 3 otteniamo:

d = Y0 dove Y0, Y1, Y2, Y3 sono i corrispettivi valori del polinomio dando ad X i valori 0, 1, 2, 3.

a + b + c + d = Y1

a x 8 + b x 4 + c x 2 + d = Y2

a x 27 + b x 9 + c x 3 + d = Y3

Otteniamo così un sistema di G + 1 equazioni lineari che risolto fornisce i coefficienti del polinomio risultante.

La soluzione del sistema viene ottenuta inserendo i valori in una matrice quadrata, che ci aiuterà a giungere alla soluzione mediante sostituzioni successive. La spiegazione del procedimento utilizzato richiederebbe una trattazione a parte e una certa conoscenza dell'algebra delle matrici.

Una volta risolto il sistema lineare, il programma visualizza i coefficienti cercati, partendo da quelli di grado maggiore.

Successivamente passa alla ricerca degli zeri del polinomio; data la lentezza di questa parte di programma, si è limitata la ricerca tra − 50 e 50, limiti stabiliti alle linee 800 e 835. La precisione è di 4 cifre dopo lo zero, servendoci della seguente formula: INT (X x 10000 + 0.5)/10000. Al posto di 10000 si può inserire 100000 per aumentare la precisione delle soluzioni, a scapito però della velocità di elaborazione. L'algoritmo che regola il processo di ricerca delle soluzioni consiste nell'inserire nella X valori in sequenza con uno step di .4 (vedi linea 810) e valutare in corrispondenza il valore del polinomio; se questo valore passa da positivo a negativo, o viceversa, significa che in quell'intorno vi è uno zero del polinomio, cioè tra il valore precedente e l'ultimo il polinomio si annulla. A questo punto si riduce lo step e si torna al punto precedente; si applica poi lo stesso principio riducendo sempre più l'intorno nel quale il polinomio si annulla, fino ad ottenere un risultato con la precisione desiderata.

Infine si studia il segno del polinomio, cioè se ne esegue la disequazione.

Lo studio viene effettuato inserendo nella X valori maggiori e minori di ogni soluzione precedentemente ottenuta e studiando i risultati.

Per le parti positive e negative si usano le abbreviazioni P e N.

Il risultato viene visualizzato nel modo seguente:

PnNm dove n, m sono termini numerici

e si deve interpretare così: il polinomio è positivo per X > n, è negativo per m < X < n.

Per rendersi meglio conto del funzionamento del programma proviamo ad inserire il seguente polinomio (vedi figura 1):

X x X x (X + 1/2) + (X − 3) x (X + 2) − (X + 4) x (X + 4) + 35/2

I coefficienti del polinomio risultante saranno:

1 0.5 −9 −4.5

Quindi il polinomio è: X ^ 3 + 0.5 x X ^ 2 − 9 x X − 4.5

Gli zeri sono: −3 −0.5 3

Il segno è: P 3 N − 0.5

P −3

per X < −3 N

Che corrisponde a:

positivo per X > 3, negativo per 3 < X < − 0.5, positivo per − 0.5 < X < − 3 infine negativo per X < − 3.

Calcolo letterale di polinomi

80-140	Calcolo del grado del polinomio.
150-280	Calcolo dei coefficienti del sistema lineare.
290-565	Risoluzione del sistema lineare.
570-695	Valutazione dei coefficienti del polinomio e stampa risultati.
700	Stampa messaggio per polinomio degenere in un termine numerico.
720-760	Risoluzione dei polinomi di I° grado e stampa risultati.
780-930	Ricerca degli zeri del polinomio.
940-970	Stampa risultati.
980-1060	Valutazione del segno del polinomio.
1070-1160	Stampa risultati.
1200-1270	Riordinamento polinomio iniziale con spostamento delle quantità a destra del segno di uguale.

Figura 3. *Elenco delle funzioni svolte dai vari blocchi del programma.*

Z (IO)	Vettore per gli zeri del polinomio.
T (G)	Vettore per i valori del polinomio in corrispondenza di altrettanti valori della X.
R (G)	Vettore per i coefficienti del polinomio risultante.
C (G, G)	Coefficienti del sistema lineare.
A$	Stringa per il polinomio.
G	Grado del polinomio.
R	Termine noto del polinomio risultante.
ST	Step per la ricerca degli zeri.
NZ	Numero delle soluzioni trovate.
X	Valori della X del polinomio.
F	Valori del risultato del polinomio.
I, K, L, J, W, F1, X1	Variabili di controllo.

Figura 4. *Lista delle variabili impiegate nel programma.*

Il quadro video corrispondente è mostrato nella figura 7.
Una puntualizzazione che dobbiamo fare riguarda la differenza, per un personal computer, tra un numero elevato a potenza e un numero moltiplicato tante volte per se stesso quanto è il valore della potenza richiesta. Matematicamente le due affermazioni hanno lo stesso significato, ma un calcolatore, in genere, non accetta numeri negativi con la prima, mentre li accetta con la seconda.
Provate ad esempio:
(− 2)ˆ 3 dà errore, mentre: (− 2) x (− 2) x (− 2) dà − 8, cioè il giusto valore.
La stessa differenza sussiste tra Xˆ 3 e X x X x X; quindi al fine di evitare interruzioni, soprattutto durante la ricerca degli zeri del polinomio (dove si usano valori negativi per la X), è consigliabile usare sempre la seconda formula.
Potete provare ad inserire polinomi più complicati:
(X x X x X − 5 x X + 7) x (X − 2) + (X x X − 4 x X − 6) x (7 x X x X + 5) − 9
oppure:
(− X x X + 8/3) x (X − 5) x (3 x X + 6) + X x X x (3 x X x X − 1) = 4 x X x X + (X + 3/2) x (X − 5)
vedi figura 3.

Modifiche proposte

Chi volesse lavorare con polinomi di grado superiore al quinto deve diminuire il valore alla linea 90, altrimenti si ottengono dei valori superiori alla capacità del computer. Il valore è dato dal numero massimo manipolato dal calcolatore diviso il grado massimo del polinomio.
Si possono aumentare i parametri di arrotondamento alle linee 575 e 625 nel modo già indicato durante l'illustrazione dell'algoritmo di ricerca degli zeri.
Infine, a chi volesse ricercare le soluzioni di polinomi di grado elevato consigliamo di diminuire il valore dello step alla linea 810.
Comunque, i parametri usati sono quelli ottimali per polinomi fino al quinto grado.

Conversioni

Il programma è stato scritto su uno ZX81 e ciò dimostra che anche macchine semplici possono eseguire programmi sofisticati. Per inserirlo in uno Spectrum basta non scrivere le linee 85, 565, 765, 785, 943, che riguardano la velocità di esecuzione (FAST o SLOW).
La conversione per altri personal è più laboriosa, poiché solo il BASIC Sinclair è dotato dell'utilissima funzione VAL (anche gli altri computer hanno la stessa istruzione, che però non fa la stessa cosa).
Ricordiamo inoltre che dopo le funzioni come ABS o SGN molti computer richiedono le parentesi e, anche se è superfluo, facciamo notare che i LET possono essere omessi.
Fatte queste precisazioni, vediamo come sostituire i VAL; le soluzioni possono essere varie: se il vostro personal è dotato del comando DEF FN al posto dell'input di linea 70 potete scrivere:
70 DEF FN A(X) = ; ... polinomio
Vanno poi tolte le linee 72, 73 e 75; infatti, non lavorando su stringhe perdono senso; non deve quindi essere scritta la subroutine che parte da linea 1200. Poi, in tutto il programma, al posto di VAL A$ si deve scrivere: FN A(X).
La sintassi di DEF FN qui presentata è quella dell'Apple-BASIC. Ricordiamo infine che i due asterischi alla linea 260 corrispondono all'elevamento a potenza. ■

Calcolo letterale di polinomi

Listato 1. Listato del programma per il calcolo letterale su polinomi in X. Accetta polinomi fino al quinto grado. Per polinomi di grado superiore occorre apportare le modifiche descritte nell'articolo. Il programma gira con 16 Kbyte di memoria, ma compattandolo dovrebbe funzionare anche solo con 4 Kbyte.

```
  10 REM ************************
  20 REM *  CALCOLO LETTERALE   *
  30 REM *   SU POLINOMI IN X   *
  40 REM ************************
  50 REM
  60 PRINT "INSERISCI IL POLINOM
IO"
  65 PRINT
  70 INPUT A$
  72 PRINT A$
  73 PRINT
  75 GOSUB 1200
  80 REM CALCOLO GRADO
  83 DIM Z(10)
  85 FAST
  90 LET X=1E7
 100 LET G=0
 110 LET F=VAL A$/X**G
 120 LET G=G+1
 130 IF ABS F>100 THEN GOTO 110
 140 LET G=G-1
 150 REM CALCOLO COEFFICENTI
 160 LET X=0
 170 LET R=VAL A$
 180 IF G=0 THEN GOTO 700
 190 DIM T(G)
 200 FOR X=1 TO G
 210 LET T(X)=VAL A$-R
 220 NEXT X
 225 IF G=1 THEN GOTO 720
 230 DIM C(G,G)
 240 FOR I=1 TO G
 250 FOR F=1 TO G
 260 LET C(I,F)=I**F
 270 NEXT F
 280 NEXT I
 290 REM RISOLUZIONE SISTEMA
 300 DIM R(G)
 310 FOR X=1 TO G-1
 320 LET I=K+1
 330 LET L=K
 340 IF ABS C(I,K)>ABS C(L,K) TH
EN LET L=I
 350 IF I>=G THEN GOTO 370
 355 LET I=I+1
 360 GOTO 340
 370 IF L=K THEN GOTO 460
 380 FOR J=K TO G
 390 LET W=C(K,J)
 400 LET C(K,J)=C(L,J)
 410 LET C(L,J)=W
 420 NEXT J
 430 LET W=T(K)
 440 LET T(K)=T(L)
 450 LET T(L)=W
 460 LET I=K+1
 470 LET W=C(I,K)/C(K,K)
 480 LET C(I,K)=0
 490 FOR J=K+1 TO G
 500 LET C(I,J)=C(I,J)-W*C(K,J)
 510 NEXT J
 520 LET T(I)=T(I)-W*T(K)
 530 IF I>=G THEN GOTO 560
 540 LET I=I+1
 550 GOTO 470
 560 NEXT K
 565 SLOW
 570 LET R(G)=T(G)/C(G,G)
 575 LET R(G)=INT (R(G)*10000+.5
)/10000
 580 FOR I=G-1 TO 1 STEP -1
 590 LET W=0
 600 FOR J=I+1 TO G
 610 LET W=W+C(I,J)*R(J)
 620 LET R(I)=(T(I)-W)/C(I,I)
 625 LET R(I)=INT (R(I)*10000+.5
)/10000
 630 NEXT J
 640 NEXT I
 645 PRINT "IL POLINOMIO E' DI ";
G;"' GRADO"
 646 PRINT "I COEFFICENTI SONO:"
 650 FOR I=G TO 1 STEP -1
 660 PRINT R(I);"  ";
 670 NEXT I
 680 PRINT R
 690 PRINT "  PER CONTINUARE PRE
MI N/L"
 695 INPUT U$
 699 GOTO 780
 700 PRINT "IL POLINOMIO DEGENER
A IN UN      TERMINE NUMERICO =";
R
 710 STOP
 720 PRINT "IL POLINOMIO E' DI 1'
GRADO"
 725 PRINT "I COEFFICENTI SONO:"
 730 PRINT T(1);"   ";R
 740 PRINT "LO ZERO DEL POLINOMI
O =";-R/T(1)
 750 LET NZ=1
 760 LET Z(1)=-R/T(1)
 765 SLOW
 770 GOTO 990
 780 REM ZERI POLINOMIO
 785 FAST
 790 LET NZ=0
 800 LET X1=-50
 810 LET ST=.4
 820 LET X=X1
 830 LET F1=VAL A$
 835 IF X>50 THEN GOTO 940
 840 LET X=X+ST
 845 LET X1=X
 850 LET F=VAL A$
 855 IF ABS (F-F1)<1E-5 OR ABS F
<1E-5 THEN GOTO 900
 860 IF F*F1>0 THEN GOTO 830
 870 LET X1=X1-ST
 880 LET ST=ST/2
 890 GOTO 820
 900 LET NZ=NZ+1
 910 LET Z(NZ)=INT (X*10000+.5)/
10000
 920 LET ST=.4
 925 LET X1=X1+.01
 930 GOTO 820
 940 PRINT
 943 SLOW
 944 IF NZ=0 THEN GOTO 1120
 945 PRINT "GLI ZERI DEL POLINOM
IO SONO:"
 950 FOR I=1 TO NZ
 960 PRINT Z(I),
 970 NEXT I
 980 REM SEGNO DEL POLINOMIO
 990 PRINT
1000 PRINT "IL SEGNO DEL POLINOM
IO E' "
1010 FOR I=NZ TO 1 STEP -1
1020 LET X=Z(I)+.01
1030 IF SGN VAL A$=1 THEN PRINT
"P ";Z(I),
1040 IF SGN VAL A$=-1 THEN PRINT
 "N ";Z(I),
1050 NEXT I
1060 LET X=X-.2
1070 PRINT
1080 PRINT "INFINE PER X<";Z(1);
"  ";
1090 IF SGN VAL A$=1 THEN PRINT
"P"
1100 IF SGN VAL A$=-1 THEN PRINT
 "N"
1110 STOP
1120 PRINT "IL POLINOMIO NON AMM
ETTE ZERI     REALI"
1130 PRINT "E' SEMPRE ";
1140 IF R>0 THEN PRINT "POSITIVO
"
1150 IF R<0 THEN PRINT "NEGATIVO
"
1160 STOP
1200 REM RIORDINAMENTO
1210 LET I=2
1220 IF I=LEN A$-1 THEN RETURN
1230 IF A$(I)="=" THEN GOTO 1260
1240 LET I=I+1
1250 GOTO 1220
1260 LET A$=A$( TO I-1)+"-("+A$(
I+1 TO )+")"
1270 RETURN
```

SOFTWARE !!

DISTRIBUITO DA

REBIT COMPUTER

ardac
ABRASCO — A TOUCH OF MAGIC
A&F software
ANIROG
ATARISOFT
Audiogenic
BUG-BYTE SOFTWARE
CDS Micro Systems
CREATIVE SOFTWARE
cp software
Datasoft®
dk'tronics
JCE
EPYX COMPUTER SOFTWARE
FIRST STAR SOFTWARE INC
HISOFT PASCAL DEVPAC
HES
HesWare
commodore
HEWSON CULSANT
Imagine
INTERCEPTOR MICRO'S ©
LLAMASOFT!!
MIKRO-GEN
MELBOURNE
ocean
PICTURESQUE
OCP
Protek
sinclair
PSION
POSTERN
QUICKSILVA
©RABBIT SOFTWARE
ROMIK
SUPERSOFT
SOFTEK MASTERS OF THE GAME
SUNSHINE
THORN EMI VIDEO
ULTIMATE PLAY THE GAME



UNA VASTA GAMMA DI PROGRAMMI, DI GIOCHI E DI UTILITÁ

DISTRIBUITI DA:

SOFTBANK DISTRIBUTION

DIVISION OF G.B.C.

Rivisitiamo insieme il Programmer's Tool Kit

La seconda versione del nostro popolare programma

di *Alessandro Guida*

Ai lettori di Personal Software

L'articolo che state per leggere si è reso necessario visto l'enorme successo riscosso dall'ormai famoso "Programmer's Tool Kit" apparso sui numeri 12/13 e 14 di **Personal Software**, *e visto che non posso rispondere privatamente ad ognuno di voi.*
Lo scopo è sia di mettere in grado tutti quanti di far funzionare tale programma sia di apportarvi alcune migliorie.

Per chi non avesse avuto modo di leggere i suddetti numeri di Personal Software, ricordiamo che il Programmer's Tool Kit è una raccolta di routine in linguaggio macchina che aggiungono al BASIC del 64 ben 11 nuovi comandi, rivolti a semplificare il lavoro di stesura e debug dei programmi.
L'articolo procederà per gradi, cercando di risolvere i problemi più frequentemente da voi incontrati.
È consigliabile non saltare alcun passo, se si vuole giungere alla fine con il programma funzionante.

Infine, una precisazione. Noi faremo sempre riferimento al primo listato apparso (quello in BASIC). La seconda parte dell'articolo, con il suo chilometrico disassemblato (tutto commentato!), aveva solamente un significato didattico. Era nostra intenzione, infatti, fornire al lettore anche la spiegazione del lavoro effettuato e dei vari problemi incontrati, con relative soluzioni. Infine, in questa maniera veniva lasciata la possibilità, ai più esperti, di modificare il tutto secondo le proprie esigenze.
Purtroppo, per poter aggiungere i commenti al disassemblato, è stato necessario riscrivere completamente il disassemblato a mano. Questo ha portato, inevitabilmente, a qualche errore che poteva essere, comunque, facilmente corretto, confrontandolo con un disassemblato del Tool Kit ottenuto dal programma in BASIC.

Dopo questo chiarimento, voglio ringraziare tutte quelle persone che con molta gentilezza mi hanno fatto notare alcune anomalie e che si sono sforzate di trovarvi una soluzione.

Vi auguro buon lavoro, ricordandovi che ben presto vi sarà un P.T.K. 2 migliorato e con altri nuovi comandi.

1. Il problema maggiore

Il primo passo da compiere è quello di mettere il programma in grado di funzionare. Frequentemente, infatti, chi ha digitato il programma si è trovato di fronte ad un misterioso ILLEGAL QUANTITY ERROR IN ... o al blocco totale del sistema. Questi guai sono dovuti in parte ad un errore (l'unico!) del listato. Era misteriosamente scomparsa una linea, che era stata prontamente riportata nel numero successivo (pag. 75) e che ripetiamo ancora:

1305 DATA A9, 2C, 2O, D2, FF, A5, FE

Inutile dire che senza questa linea il 64 va inesorabilmente incontro ad un blocco totale.

Il resto degli insuccessi, decretati dal messaggio di errore visto prima, sono dovuti ad errori di battitura degli innumerevoli DATA. Naturalmente, il messaggio riporta il numero della linea in cui si è cercato di fare una POKE con un numero maggiore di 255 (la linea 90) o di quelle in cui si cerca di determinare l'ASC di una stringa nulla, ma questo non ci è di alcun aiuto. Per conoscere invece la linea di DATA che ha generato l'errore è sufficiente digitare in modo diretto:

PRINT PEEK (63) + PEEK (64) ★ 256

Il numero ottenuto sarà proprio quello della linea incriminata.

Infine, un piccolo numero di lettori ha avuto problemi nell'interpretare i caratteri grafici che appaiono nelle linee comprese tra la 25 e la 38. In realtà, questi non sono altro che lettere in maiuscolo formanti una maschera video, che richiama la sintassi dei comandi. In ogni caso, riportiamo nuovamente il listato corretto e nel set maiuscolo/minuscolo.

2. Alla ricerca del BUG

Può darsi, però, che il programma non dia alcuna segnalazione di errore e arrivi alla fine senza che il cursore ricompaia. Oppure che una parte delle nuove istruzioni non funzioni. Anche in questi casi il problema è nei DATA. È sufficiente che uno solo dei 1694 dati presenti sia errato perché appaiano strani risultati.
Per ovviare a questi inconvenienti, nel listato 2 troverete un programma dal simbolico nome di: "CONTROLLER".
Dopo aver caricato il vostro Tool Kit (la versione in BASIC ovviamente) aggiungetevi le linee del pro-

Rivisitiamo insieme il Programmer's Tool Kit

gramma 2. Queste partono da 10000 e finiscono quindi in coda al PTK. Finito di digitarle, date RUN 10000. Se vi sono dei DATA errati, il computer vi segnalerà il blocco di linee in cui gli errori sono compresi. Effettuate le correzioni si darà ancora RUN 10000, finché non si avranno più segnalazioni di errore.
A questo punto il vostro Tool Kit è in ordine e potete salvarlo su nastro, prima di dare il RUN e dopo aver cancellato le linee da 10000 in poi. Naturalmente, si deve prestare la massima attenzione nel digitare le linee aggiuntive poiché non ci sarà alcun programma per controllare il CONTROLLER!

3. L'istruzione MERGE

Se siete giunti a questo punto con il Tool Kit funzionante, si può proseguire con i perfezionamenti del programma.
L'istruzione Merge ha il difetto di funzionare solo con il registratore e solo se non si specifica il numero di periferica. Diversamente si ha l'errore DEVICE NOT PRESENT, anche se il disco o il nastro sono normalmente collegati.
Questo dipende dal fatto che il numero in ASCII non viene convertito in binario prima della apertura del canale con la periferica, per cui il computer, di fronte alla richiesta del dispositivo 49 (il codice dell'uno in ASCII) o del 56, non può far altro che segnalare "periferica assente".
Per rimediare a questo problema va corretta la linea 1002.
Vanno sostituiti il 16mo e il 17mo dato (B0, E9) con 29, 0F.
Nel listato 1 tale linea appare già corretta.

4. L'istruzione KEY

Per l'istruzione KEY molti hanno lamentato il fatto che con la pressione dei tasti STOP e RESTORE i tasti di funzione vengono disabilitati. In realtà, questo è opportuno sia così, poiché può essere necessario riportare i tasti di funzione allo stato normale, per eseguire altri programmi che li utilizzano.
Apporteremo, però, una piccola modifica al Tool Kit, per fare in modo che i tasti di funzione vengano riabilitati digitando il comando KEY.
a) Nell'ultima linea di DATA, la 2030, vanno tolti l'asterisco e la virgola che lo precede.
b) Bisogna aggiungere la linea 2040 come segue:
2040 DATA 78, A9, 9A, 8D, 15, 03, A9, 90, 8D, 14, 03, 58, 4C, 73, 00, ★
c) Va corretta la linea 1230 come segue:
1230 DATA 20, 68, 9F, F0, 3B

5. Un difetto comune a MERGE, TRON, AUTO

Tutti questi comandi presentavano dei problemi nel gestire i numeri superiori a 32768. Questo fatto è dovuto alla routine in linguaggio macchina, che si occupa di trasformare un numero in notazione intera (2 byte) in uno a virgola mobile.
La routine utilizzata ($B391) trasforma tali numeri, considerando il bit 7 del byte più significativo come il segno, di conseguenza superando il valore di 32768 si passa ai numeri negativi. Si può in ogni caso porre rimedio con le modifiche sottoindicate:
a) Modificare le seguenti due linee così:
15 POKE 56, 152: POKE 55, 240: CLR
45 IF A$ = "★" THEN F = 1 : IN = 39152 : GOTO 40
b) Aggiungere la linea:
980 DATA 00, 00, A2, 00, 86, 0D, 85, 62, 84, 63, A2, 90, 38, 4C, 49, BC
c) Modificare i seguenti DATA:
Linea 1010, sostituire l'undicesimo e 12mo dato (91, B3) con 2F, 99.
Linea 1645, sostituire l'ottavo e il nono dato (91, B3) con 2F, 90.
Linea 1780, sostituire gli ultimi due dati come per le due linee precedenti.
Se avete seguito passo per passo fin qui, il vostro programma dovrebbe essere identico a quello del listato 1.

6. L'ultimo miglioramento

A questo punto il programma è a punto e perfettamente funzionante. L'unico neo rimasto è quello dovuto alla lentezza in fase di caricamento ed inizializzazione del tutto. Ma è possibile risolvere anche questo, utilizzando un caricatore e registrando il Tool Kit direttamente dalla zona di memoria da esso occupata.
a) Caricare normalmente il Tool Kit e dare il RUN per attivarlo.
b) Digitare il programma loader del listato 3 e salvarlo su disco o su nastro con il nome "PTK LOADER". Se utilizzate il nastro, ricordate che questo programma deve essere il primo ad essere registrato e che nella linea 40 va cambiato il numero della periferica.
c) Digitate ora in modo diretto:
LOAD "PTK", 8, se utilizzate il disco 0,
LOAD "PTK", 1, se usate il registratore.
Battete RETURN e subito dopo premete il tasto di STOP.

Rivisitiamo insieme il Programmer's Tool Kit

Listato 1. *Listato del Programmer's Tool Kit con le modifiche già apportate.*

```
2 rem **********************************
3 rem *                                *
4 rem *  Programmer's tool kit  v2    *
5 rem *                                * 6 rem **********************************
10 poke56,152:poke55,240:clr:rem topmem
20 in=828
25 printchr$(14)"██      ***PROGRAMMER'S   TOOL  KIT***      "
27 print"SINTASSI COMANDI:"
28 print"────────────────"
30 print"HELP":print"█AUTO[in,][step]":print"█FIND,["chr$(34)"stringa"chr$(34)"]
";
32 print"/[variabile/keyword]"
34 print"█DUMP":print"█REN[in,][step]":print"█TRON[,STEP]/[,S]"
35 print"█TROFF":print"█DELETE[in-fine]/[in-]/[-fine]"
36 print"█KEY[n,][stringa][←]":print"█MERGE[filename][,dev]"
38 print"██attendere, prego!█";
40 reada$:ifa$="*"andf=1then100
45 ifa$="*"thenf=1:in=39152:goto40
50 a1$=left$(a$,1)
60 gosub500:n=rs*16
70 a1$=right$(a$,1)
80 gosub500:n=n+rs
90 pokein,n:in=in+1:goto40
100 print"█PROGRAMMER'S TOOL KIT ----█ATTIVATO"
110 sys(828):new
500 ifasc(a1$)<58thenrs=asc(a1$)-48:return
510 rs=asc(a1$)-55:return
900 data78,a9,9f,85,fc,a9,80,85,fb,a9,50,8d,02,03,a9,9a,8d,03,03
910 dataa9,00,a0,7f,91,fb,88,10,fb,a9,9a
920 data8d,15,03,a9,90,8d,14,03,a9,8f,8d,00,03,a9,9c,8d,01,03,58,60,ea,*
950 data48,45,4c,50,9c,c6,41,55,54,4f,9d,61,46,49,4e,44,9c,12,44,55,4d,50,9b,56
960 data52,45,4e,9f,12,54,52,4f,9d,c0,4d,45,52,47,45,99,3d,10,10,10,10,a0,a0
970 data44,45,4c,45,54,45,9e,af,4b,45,59,9a,d8,00
980 data00,00,a2,00,86,0d,85,62,84,63,a2,90,38,4c,49,bc
1000 data68,68,a2,00,20,73,00,f0,1f,c9,22,f0,07,c9,2c,f0,12,4c,08,af,20,73,00
1002 dataf0,f8,c9,22,f0,e7,9d,00,02,e8,d0,f1,20,73,00,29,0f,85,fe,8a,a2,00,a0
1004 data02,20,bd,ff,a6,fe,d0,01,e8,a0,00,a9,01,20,ba,ff,a6,2d,86,f7,a4,2e,c8
1006 data84,f8,a9,00,20,d5,ff,90,04,aa,6c,00,03,86,f9,84,fa,a9,9d,8d,02,03,a9
1008 data99,8d,03,03,a0,01,84,0f,b1,f7,d0,03,4c,05,9f,c8,b1,f7,85,14,c8,b1,f7
1010 data85,15,c8,84,fe,a4,14,a5,15,20,2f,99,20,dd,bd,a2,00,bd,01,01,f0,06,9d
1012 data00,02,e8,d0,f5,a4,fe,d0,0a,c9,22,d0,06,a5,0f,49,ff,85,0f,b1,f7,f0,3f
1014 data30,07,9d,00,02,c8,e8,d0,e9,c9,ff,f0,f5,24,0f,30,f1,38,e9,7f,86,49,84
1016 datafe,aa,a0,ff,ca,f0,08,c8,b9,9e,a0,10,fa,30,f5,a6,49,c8,b9,9e,a0,30,06
1018 data9d,00,02,e8,d0,f4,29,7f,9d,00,02,a4,fe,c8,e8,d0,bd,9d,00,02,98,38,65
1020 dataf7,85,f7,90,02,e6,f8,a2,ff,a0,01,4c,86,a4,00,00,00,00,00,00,4b,45,59
1022 data00,00,00
1050 data4c,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
1060 data20,60,a5,86,7a,84,7b,a0,00,a2,00,b1,37,d0,03,4c,8a,a4,30,12,dd,00,02
1070 datad0,04,c8,e8,d0,ee,c8,b1,37,10,fb,c8,c8,d0,e3,8d,3f,9a,c8,b1,37
1080 data8d,3e,9a,a9,ff,85,3a,18,ca,86,7a,e6,7b,20,3d,9a,4c,74,a4,ea
1160 dataa5,c5,c5,02,f0,0d,85,02,38,e9,03,30,06,f0,07,c9,04,30,05,4c,31,ea
1170 dataa9,04,a8,88,98,0a,0a,0a,0a,0a,85,fd,ad,8d,02,29,01,f0,07
1180 data18,a5,fd,69,10,85,fd,a4,fd,a2,00
1190 datab1,fb,f0,07,9d,77,02,c8,e8,d0,f5
1200 data86,c6,68,a8,68,aa,68,40,ea
1230 data20,68,9f,f0,3b
1240 data90,03,4c,08,af,85,fe,c6,fe,a9,2c
1250 dataa0,01,20,01,af,a5,fe,0a,0a,0a,0a
1260 dataa8,a2,0a,20,73,00,90,0f,f0,18,c9,5f
1270 dataf0,07,20,13,b1,b0,04,90,d7,a9,0d
1280 data91,fb,c8,ca,d0,e6,a2,17,6c,00,03
```

Rivisitiamo insieme il Programmer's Tool Kit

Seguito listato 1.

```
1290 data91,fb,60,a2,00,86,fe,a9,37,a0,9a
1300 data20,1e,ab,a5,fe,18,69,31,20,d2,ff
1305 dataa9,2c,20,d2,ff,a5,fe
1310 data0a,0a,0a,0a,a8,b1,fb,f0,0c,c9,0d
1320 datad0,02,a9,5f,20,d2,ff,c8,d0,f0,a9,0d
1330 data20,d2,ff,e6,fe,a5,fe,c9,08,d0,c8,60
1350 dataea,20,73,00,f0,03,4c,08,af,a5,2d,85,fe,a5,2e,85,ff,a0,00,a9,01,85,0d
1360 datab1,fe,85,46,30,02,84,0d,c8,b1,fe,85,45,30,04,a9,ff,85,0d,a2,02,b5,44
1370 data29,7f,20,d2,ff,ca,d0,f6,a5,0d,30,06,18,69,24,20,d2,ff,a9,3d,20,d2,ff
1380 dataa5,0d,f0,4c,a5,0d,30,36,c8,b1,fe,48,c8,b1,fe,a8,68,20,91,b3,20,dd,bd
1390 dataa9,00,a0,01,20,1e,ab,a9,0d,20,d2,ff,18,a5,fe,69,07,85,fe,a5,ff,69,00
1400 data85,ff,c5,30,f0,03,4c,66,9b,a5,fe,c5,2f,90,f7,60
1410 data18,c8,98,65,fe,aa,a5,ff,69,00,a8,8a,20,a2,bb,4c,ae,9b,a9,22,20,d2,ff
1420 datac8,b1,fe,f0,16,aa,c8,b1,fe,85,69,c8,b1,fe,85,6a,a0,00,b1,69,20,d2,ff
1430 datac8,ca,d0,f7,a9,22,20,d2,ff,4c,b8,9b
1440 data20,73,00,c9,2c,f0,03,4c,08,af,a0,00,20,73,00,f0,f6,a5,7a,c9,05,d0,f0
1445 dataa9,00,85,7a,20,7c,a5,a5,2b,85,f9,a5,2c,85,fa
1450 dataa9,ff,8d,4f,9a,a2,00,a0,01,b1,f9,d0,06,a9,00,8d,4f,9a,60,c8,a5,f9
1460 data85,5f,a5,fa,85,60,b1,f9,85,14,c8,b1,f9,85,15,c8,bd,05,02,f0,17
1465 datac9,22,d0,03,e8,d0,f4,b1,f9
1470 dataf0,11,dd,05,02,d0,03,e8,d0,e7,a2,00,f0,e3,a9,00,20,bd,a6,a0,00,b1,f9
1480 dataaa,c8,b1,f9,85,fa,8a,85,f9,18,90,ae
1500 data8a,10,09,ad,4f,9a,f0,01,60,4c,74,a4,aa,a5,7a,8d,42,9a,a5,7b,8d,43,9a
1510 dataa5,39,8d,40,9a,a5,3a,8d,41,9a,8a,4c,3a,a4
1550 dataaa,ce,4e,9a,f0,04,8a,4c,1a,a7,a9,3f,20,d2,ff,18,90,f4
1560 data20,73,00,f0,03,4c,08,af,ad,40,9a,85,14,ad,41,9a,c9,ff,d0,01,60,85,15
1570 dataa9,ff,8d,4f,9a,a9,b4,8d,06,03,a9,9c,8d,07,03,20,13,a6,38,ad,42,9a
1580 datae5,5f,38,e9,04,8d,4e,9a,20,bd,a6,a9,1a,8d,06,03,a9,a7,8d,07,03
1590 dataa9,00,8d,4f,9a,60
1600 dataa0,00,8c,45,9a,8c,47,9a,a2,0a,20,73,00,d0,07,8e,44,9a,8e,46,9a,60,90,06
1610 data84,15,86,14,d0,03,20,6b,a9,c9,2c,f0,07,c9,2d,f0,03,4c,08,af,a5,14
1620 data8d,44,9a,a5,15,8d,45,9a,20,73,00,f0,07,b0,ec,20,6b,a9,b0,06,85,15,a9,0a
1630 data85,14,a5,15,8d,47,9a,a5,14,8d,46,9a,60
1635 data68,68,20,0e,9d,ad,46,9a,0d,47,9a,f0,0d,a9,88
1640 data8d,02,03,a9,9d,8d,03,03,6c,02,03,a9,50,8d,02,03,a9,9a,8d,03,03,4c,74,a4
1645 dataac,44,9a,ad,45,9a,20,2f,99,20,dd,bd,a0,01
1647 datab9,00,01,f0,06,99,76,02,c8,d0,f5,a9,20,99,76,02,84,c6
1650 data18,ad,46,9a,6d,44,9a,8d,44,9a,ad,47,9a
1660 data6d,45,9a,8d,45,9a,b0,be,4c,50,9a
1700 dataa9,00,8d,49,9a,20,73,00,c9,46,f0,41,c9,4e,f0,03,4c,08,af,20,73,00
1710 dataf0,10,c9,2c,d0,f4,20,73,00,c9,53,d0,ed,a9,01,8d,49,9a,a0,22,a2,06,a9,a0
1720 data99,00,04,ad,86,02,99,00,d8,c8,ca,d0,f1,98,18,69,22,a8,90,e8,a9,18
1730 data8d,08,03,a9,9e,8d,09,03,60,a9,e4,8d,08,03,a9,a7,8d,09,03,60
1740 dataa5,3a,c9,ff,d0,03,4c,e4,a7,a5,39,a4,3a,cd,4a,9a,d0,05,cc,4b,9a,f0,ef
1750 data8d,4a,9a,8c,4b,9a,ad,49,9a,f0,07,ea,a5,c5,c9,40,f0,f9,a9,4a,85,52
1760 dataa9,04,85,51,85,53,a9,22,85,50,a0,05,b1,52,91,50,88,10,f9,18,a5,50
1770 data69,28,85,50,a5,52,69,28,85,52,90,e8,a9,22,85,50,a9,d8,85,51,ad,86,02
1780 dataa0,05,91,50,88,10,fb,18,a5,50,69,28,85,50,90,ed,a5,3a,a4,39,20,2f,99
1790 data20,dd,bd,a9,a3,8d,ea,04,a0,01,b9,00,01,f0,08,09,80,99,ea,04,c8,d0,f3
1800 dataa9,a0,c0,07,f0,06,99,ea,04,c8,d0,f6,4c,e4,a7
1900 data68,68,20,0e,9d,a2,04,bd,43,9a,95,f6,ca,d0,f8,a5,fa,d0,0a,a5,f9,c9,0a
1910 datad0,04,a9,ff,85,fa,a5,f7,85,14,a5,f8,85,15,20,13,a6,a9,e1,8d,02,03
1920 dataa9,9e,8d,03,03,a9,00,8d,00,02,a0,01,b1,5f,f0,19,c8,c8,b1,5f,c5,fa
1930 dataf0,04,b0,0f,90,09,88,b1,5f,c5,f9,f0,02,b0,04,38,20,a9,a4,a9,9a
1940 data8d,03,03,a9,50,8d,02,03,4c,74,a4
2000 data20,0e,9d,ad,47,9a,f0,05,a2,0e,6c,00,03,ad,46,9a,f0,f6,a5,2b,85,f7
2010 dataa5,2c,85,f8,a0,01,b1,f7,d0,01,60,85,fa,88,b1,f7,85,f9,c8,c8,ad,44,9a
2020 data91,f7,c8,ad,45,9a,91,f7,a5,f9,85,f7,a5,fa,85,f8,18,ad,44,9a,6d,46,9a
2030 data8d,44,9a,ad,45,9a,69,00,8d,45,9a,90,c9,a2,0f,6c,00,03
2040 data78,a9,9a,8d,15,03,a9,90,8d,14,03,58,4c,73,00,*
```

PERSONAL SOFTWARE

Rivisitiamo insieme il Programmer's Tool Kit

Listato 2. *Programma controller, necessario per verificare i DATA del Programmer's Tool Kit.*

```
10000 RESTORE
10010 READA$:IFA$<>"*"THEN10010
10020 READA$:IFA$<>"*"THEN10020
10030 FORI=0TO9:FORJ=0TO2:READA(I,J):NEXTJ,I
10040 RESTORE:C=0:PRINT"■"
10050 T=0:FORI=0TOA(C,0):READA$
10060 T=T+ASC(A$)+ASC(RIGHT$(A$,1)):NEXTI
10070 IFT<>A(C,1)THENPRINT"ERRORE NELLE LINEE"A(C,2)"-"A(C+1,2)
10080 C=C+1:IFC<9THEN10050
10100 DATA51,5912,900,198,22419,950,136,15343,1012
10110 DATA162,18687,1060,228,26487,1260,237,27655,1410
10120 DATA271,30802,1560,216,24626,1710,184,21299,1900,0,0,2030
```

Listato 3. *Programma loader necessario per caricare il Tool Kit direttamente in memoria, da disco o da nastro.*

```
2 rem ***********************************
3 rem *                                 *
4 rem *  Programmer's tool kit  v2      *
5 rem *                                 *
6 rem ***********************************
10 if c=1 then 50
20 poke56,152:poke55,240:clr:c=1:rem topmem
25 printchr$(14)"■■      ***PROGRAMMER'S   TOOL  KIT***      "
27 print"SINTASSI COMANDI:"
28 print"────────────────"
30 print"HELP":print"■AUTO[in,][step]":print"■FIND,["chr$(34)"stringa"chr$(34)"]
";
32 print"/[variabile/keyword]"
34 print"■DUMP":print"■REN[in,][step]":print"■TRON[,STEP]/[,S]"
35 print"■TROFF":print"■DELETE[in-fine]/[in-]/[-fine]"
36 print"■KEY[n,][stringa][←]":print"■MERGE[filename][,dev]"
38 print"■■attendere, prego!■";
40 load"ptk",8,1
50 poke251,128:poke252,159:poke770,80:poke771,154:poke768,143:poke769,156
60 fori=0to127:poke40832+i,0:next
70 print"■Programmer's Tool Kit --- ■ATTIVATO■"
80 new
```

Lo scopo di questa manovra è di assegnare ad un file il nome "PTK" e il numero di periferica 8 (o 1).
d) Digitate, ancora in modo diretto:
POKE 139, 240 : POKE 194, 152 :
POKE 174, 0 : POKE 175, 160
Questo fissa il punto di partenza e di fine della zona di memoria da salvare.
Si darà ora SYS 62957 ed il Tool Kit verrà salvato direttamente su disco o su nastro.
Da ora in poi sarà sufficiente caricare il loader e dare il RUN, perché il Tool Kit venga caricato velocemente nella giusta parte della memoria.
Con questo ultimo perfezionamento abbiamo terminato.
A risentirci presto con il "*PTK 2*". ■

MEMORIE MAGNETICHE PER COMPUTER

RPS

RHÔNE-POULENC SYSTEMES



IL CUORE DI "RHÔNE POULENC" È LA CHIMICA

Senza la Chimica non potrebbero esistere oggi l'Industria Spaziale, la Micro Elettronica, l'Informatica.
Tra i primi gruppi chimici nel mondo, RHÔNE POULENC, ha scelto l'informatica come asse di sviluppo strategico del gruppo.

Sviluppando il suo potenziale chimico, l'esperienza dei suoi ricercatori (7000 persone nel mondo) e sfruttando dei processi di fabbricazione e di controllo esclusivi in EUROPA, messi a punto nella SILICONE VALLEY, RHÔNE POULENC SYSTEMES, è diventata in quattro anni, nel campo delle memorie magnetiche, un simbolo di sicurezza riservato ad una clientela sempre più esigente.

la sicurezza delle tecnologie più avanzate

RPS
RHÔNE-POULENC SYSTEMES

concessionari autorizzati

BRENUANI MASSIMO
Via Chiusi, 76
00139 ROMA
Tel. 06/8127665-8120727

CSS s.n.c.
Via Fra P. Sarpi, 8/A
50136 FIRENZE
Tel. 055/679630

DATAPLAN s.a.s.
Via Cassa di Risparmio, 9
39100 BOLZANO
Tel. 0471/47721

MIDA s.r.l.
Via Dietro Filippini, 1/A
37121 VERONA
Tel. 045/590505

NUOVA TECNODATA s.a.s.
Via Dalmazia, 6/B
43100 PARMA
Tel. 0521/25079

PROGRAMMA UFFICIO s.a.s.
Corso Francia, 92/A
10093 COLLEGNO (TO)
Tel. 011/4113565

RAVECO-LINE s.r.l.
Via S. G.B. De la Salle, 4
20132 MILANO
Tel. 02/2566849-2568802

SDC-EDPRINT s.r.l.
Largo Promessi Sposi, 5
20142 MILANO
Tel. 02/8435593-8466538

STUDIO SINTESI s.a.s.
Via Aldighieri, 61
44100 FERRARA
Tel. 0532/21507

TES-IN & C. s.r.l.
Via Caravaggio, 82
80126 NAPOLI
Tel. 081/643122-646752

Dal BASIC al Pascal

— Parte sesta —

La correzione dei programmi Pascal

Nella quinta parte, pubblicata sul n. 17 di **Personal Software**, abbiamo visto il primo programma tradotto in Pascal dal BASIC. Qui di seguito presentiamo il listato BASIC del programma REVERSE.

Ora passiamo ad occuparci di un argomento molto importante: la correzione dei programmi.

La correzione dei programmi Pascal avviene in due fasi.

Nella prima si eliminano tutti gli errori trovati dal compilatore nel tentativo di compilare il programma. All'inizio vi potrete irritare per quanto il compilatore sia "pignolo" riguardo i piccoli problemi di sintassi, ma più tardi vorreste che il compilatore riuscisse a trovarvi anche gli errori più sfuggenti.

Personalmente ho lavorato con cinque compilatori Pascal, e hanno tutti la stessa particolarità. Presumo che sia una conseguenza naturale del modo in cui il linguaggio è definito. Alcuni errori fanno sì che il compilatore perda la sincronizzazione. Un tale problema fa generare molti messaggi di errore senza senso nei pezzi di programma seguenti, dove non esistono errori.

La procedura giusta, dunque, consiste nel correggere il primo errore riportato. Se è uno di quelli che causano problemi di sincronizzazione, l'inconveniente sparisce non appena lo si corregge.

Listato 1. *Il programma BASIC.*

```
10 print tab(22);"REVERSE"
20 print tab(15);"CREATIVE COMPUTING"
30 print: print: print
100 print"REVERSE - UN GIOCO DI ABILITA'"
110 print
130 dim a(20)
140 rem ***** n=numero di cifre
150 n=9
160 print"vuoi le regole";
170 input a$
180 if left$(a$,1)='N' then 210
190 gosub 710
200 rem ***** crea una lista casuale da A(1) a A(N)
220 for k=1 to n
230 a(k)=k: next k
240 for k=n to 2 step -1: j=int((k-1)*rnd(0)+2)
250 t=a(k): a(k)=a(j): a(j)=t
260 next k
280 rem ***** stampa la lista e inizia il gioco
290 print: print"PARTIAMO...LA LISTA E':"
310 t=0
320 gosub 610
330 print"QUANTI NE INVERTO";
340 input r
350 if r=0 then 520
360 if r<=n then 390
370 print"OOPS! TROPPI! NE POSSO INVERTIRE ";
380 print"AL MASSIMO";n: GOTO 330
390 t=t+1
400 rem ***** inverti r n numeri e stampa la lista
410 for k=1 to int (r/2)
420 z=a(k)
430 a(k)=a(r-k+1)
440 a(r-k+1)=z
450 next k
460 gosub 610
470 rem ***** verifica se ho vinto
480 for k=1 to n
490 if a(k)<>k then 330
500 next k
510 print"HAI VINTO IN";t;"MOSSE!":print
520 print
530 print"VUOI GIOCARE ANCORA (SI O NO)";
540 input a$
550 if left$(a$,1)="S" then 210
560 print:print"O.K. SPERO TI SIA DIVERTITO"
570 goto 999
600 rem ***** subroutine che stampa la lista
610 print: for k=1 to n: print a(k);: next k
650 print: print: return
700 rem ***** subroutine che stampa le regole
705 print
710 print"QUESTO E' IL GIOCO DEL 'REVERSE'."
715 print"PER VINCERE DEVI RIORDINARE LA"
720 print"LISTA DEI NUMERI DA 1 A";n
730 print"IN ORDINE CRESCENTE DA SINISTRA"
```

Seguito listato 1.

```
735 print"A DESTRA. PER MUOVERE DEVI DIRMI"
740 print"QUANTI NUMERI (DALLA SINISTRA)"
750 print"DEVO INVERTIRE."
755 print
760 print"SE LA LISTA E'"
770 print"2 3 4 5 1 6 7 8 9"
775 print"E NE VUOI INVERTIRE 4 IL"
780 print"RISULTATO SARA':"
785 print"5 4 3 2 1 6 7 8 9"
790 print"E SE ORA NE INVERTI 5 HAI VINTO"
800 print"1 2 3 4 5 6 7 8 9"
830 print"PER FINIRE BATTI 0"
840 print: return
999 end
```

Alcuni compilatori "si confondono" talmente tanto quando c'è un errore che abortiscono invece di continuare. Altri cercano di eseguire una modifica e risincronizzarsi, in modo da poter continuare a dare delle informazioni utili, per correggere gli errori con un passaggio unico attraverso il compilatore. Questi ultimi compilatori vanno meglio perchè permettono di correggere gli errori trovati, in modo più rapido.

Le variabili non inizializzate

La seconda fase della correzione di un programma Pascal è quella di fargli fare ciò che volete che faccia, dopo che è stato compilato con successo. Ci sono alcuni trabocchetti ai quali dovete stare attenti.
C'è un punto, in particolare, che può causare dei problemi, perché qualcuno che ha programmato con il BASIC può non essersene accorto. Nel BASIC tutte le variabili usate nel programma vengono azzerate automaticamente prima che si usino. Quindi si può usare una variabile senza averle dato un valore. I programmatori che usano il BASIC possono ignorare che, per quanto io sappia, il BASIC è l'unico linguaggio di programmazione per il quale questo è possibile. Per tutti gli altri potete assumere che il valore di una variabile sia sconosciuto, finché non le avete fissato un valore per la prima volta. Un'eccezione a ciò è il linguaggio FORTH, il quale vi costringe a stabilire un valore iniziale quando definite una variabile. Lo stabilire il valore iniziale di una variabile si chiama inizializzare. La mancata inizializzazione di una variabile significa che il programma ha solo dello spazio riservato nella memoria per il valore della variabile. Quello che c'era nella memoria dal programma precedente vi rimane. Potrebbe succedere che la variabile assuma qualunque valore che le è permesso. Il risultato può essere disastroso in molti casi.

Il caso più ovvio in cui una variabile non inizializzata può causare un problema è quello in cui la variabile si usa come indice di un ciclo come ad esempio in REPEAT-UNTIL o WHILE DO. Supponiamo che N sia un indice di ciclo come sotto:

```
WHILE N<100 DO
BEGIN
    SOMMA:=SOMMA+N;
    N:=N+1;
END;
```

Se siete fortunati, N avrà un valore fra 0 e 100 e il ciclo terminerà bene, ma con la risposta sbagliata per SOMMA. Nella maggior parte dei casi, *N* partirà con un valore molto lontano dall'intervallo desiderato e il ciclo continuerà per moltissimo tempo, causando forse un "*integer overflow*" a qualche punto. Naturalmente, il ciclo dovrebbe essere preceduto da un'istruzione che fissa N := 0; o qualunque altro valore iniziale che volete.

Istruzioni sbagliate

Un altro problema difficile da rilevare nel Pascal è un punto e virgola nel posto sbagliato. Può essere collocato in modo tale da non causare un errore di sintassi, ma l'istruzione non fa più ciò che doveva. Un posto in cui un punto e virgola in più è difficile da trovare è illustrato dall'esempio seguente:

```
FOR N:=1 TO 100 DO;
BEGIN
    SOMMA:=SOMMA+N
END;
```

Naturalmente non usereste un BEGIN e un END per un'unica istruzione; in questo caso ho esemplificato. Il punto e virgola, dopo DO termina il ciclo. Nel BASIC, per fare questo, vi siete abituati a una istruzione NEXT N. Qui, il ciclò non fa niente per 100 volte e poi elabora una sola volta l'istruzione composta ... che non è non esattamente ciò che si voleva.

Potete crearvi molte difficoltà se aggiungete un punto e virgola in più a una struttura IF THEN ELSE. Tutti i sistemi Pascal che ho permettono un punto e virgola dopo l'ultima dichiarazione in un'istruzione composta, anche se il Pascal non lo richiede. Tuttavia, nessuno dei siste-

mi permette un punto e virgola dopo l'ultimo caso in un'istruzione CASE (subito prima dell'END). Non so esattamente perché, ma è così, dunque state attenti.

Dividere per vincere

Poiché il BASIC è di solito usato come interprete, è generalmente interattivo. I programmatori, a volte usano delle tecniche di correzione che sono essenzialmente dei metodi "prova e riprova". Sono come i tecnici dei primi giorni della televisione, che cambiavano i tubi finché il televisore non funzionava, oppure lo riportavano in laboratorio dove qualcuno, che ne sapeva più di loro, poteva lavorarci. È così facile cambiare un'istruzione nel BASIC, e rieseguire il programma, che questa tecnica di solito funziona. Con un compilatore, però, il passo compilativo crea dei problemi. Dobbiamo elevare al massimo la quantità di informazioni ricavata ogni volta che compiliamo ed eseguiamo il programma. Ci sono molte tecniche di programmazione di un linguaggio compilato che hanno come risultato minore sforzo nel controllare e correggere.

Guardando il programma REVERSE nelle versioni BASIC e Pascal, avete probabilmente notato che nella versione BASIC ci sono solo due subroutine. Le altre sezioni, che nella versione Pascal abbiamo scritto come procedure, sono in linea e il programma non ne soffre, almeno funzionalmente. Uno dei motivi principali per un maggior uso delle procedure e delle funzioni nel Pascal è che è più facile correggere il programma.

Avrete senz'altro sentito lo slogan "dividere per vincere". Per darvene un esempio, supponiamo che nella quinta parte aveste scritto per prima funzione RANDOM. Avendo già dichiarato le costanti e le variabili, perché non fare un programmino veloce per controllare RANDOM?

```
(PROGRAMMA PRINCIPALE)
BEGIN
   FOR J:=1 TO 10 DO
   BEGIN
      FOR K:=1 TO 10 DO WRITE (RANDOM:2);
      WRITELN
   END;
END.
```

Ora, se elaborate ciò che avete di REVERSE, ricevete dieci righe, ognuna di 10 numeri random, e vi potete convincere che RANDOM funziona bene. Quando siete soddisfatti di RANDOM basta cancellare il programma "principale" e scrivere la procedura CREALISTA.
Ora la si può controllare con un programma principale diverso.

```
BEGIN
   FOR K:=1 TO 16 DO
   BEGIN
      CREALISTA;
      FOR J:=1 TO 9 DO WRITE (LISTA[J]:2);
      WRITELN;
   END;
END.
```

A questo punto un'elaborazione di REVERSE dà (se CREALISTA non contiene errori) 16 righe di numeri da 1 a 9 in ordine casuale senza duplicati sulla stessa riga.
Potreste volere cambiare K da 16 a 1000, o qualcosa del genere, per vedere se l'elenco random non va in errore e ripete sempre un certo ordinamento (il che mi è successo su alcuni tentativi di scrittura di RANDOM).

Write, write, write

Supponete che a questo punto CREALISTA si rifiuti di funzionare bene. Senza la possibilità di controllare interattivamente, come nel BASIC, come si fa capire cosa c'è che non va? Parafrasando lo slogan di prima, "sottodividere per vincere". Inserite semplicemente alcune istruzioni WRITE per vedere ciò che succede. Faccio partire tutte le mie istruzioni WRITE di controllo dalla prima colonna, cioè senza indentarle. Il motivo è che sono più facili da vedere quando viene il momento di toglierle. Se un'istruzione non è molto ovvia, la commento con una riga di asterischi. Un po' di riflessione sui punti dove mettere le istruzioni WRITE, all'interno di una procedura che non funziona, può ridurre il numero di posti dove si deve cercare l'errore, fino a poche righe. Spesso l'errore diventa evidente quando si fanno stampare i valori delle variabili. La migliore strategia, naturalmente, è di mettere le istruzioni WRITE in punti tali nella procedura da poter vedere che le cose vanno come devono in un punto, ma un paio di righe più in là le variabili hanno dei valori imprevisti.

La tecnica "fare qualcosa che non c'entra"

Torniamo ai problemi di correzione. Cosa dovete fare quando proprio non riuscite a trovare la soluzione? Basandomi su di una lunga esperienza, il migliore consiglio che vi posso dare è smettere e fare qualcos'altro per un po'. Spesso, un errore che sembra impossibile da eliminare risulta chiaro dopo un breve

riposo. Spesso, quando ho un problema così mi viene in mente la soluzione, o altre cose da provare, quando sto cercando di fare qualcosa che non c'entra proprio.
La mente va in "*loop*" e continua a cercare sempre negli stessi posti, tantissime volte, quando si è stanchi o affaticati. Mantenete una mente aperta. Non eliminate alcuna possibilità, neanche l'ultima procedura che avete appena finito di controllare e che avete trovato "senza errori".

Usate gli aiuti del compilatore per correggere

La maggior parte dei compilatori è dotata di alcuni dispositivi di controllo. Questi possono variare da un minimo, che vi dice dov'era il programma quando è successo l'errore di elaborazione, alla possibilità di esaminare il programma passo per passo e di studiare le variabili.
Molti compilatori hanno delle opzioni che permettono di segnare i numeri di linea in modo speciale come commenti, così che, quando si corregge, il numero di linea può essere stampato insieme all'informazione sull'errore. Il mio compilatore genera un elenco identico al programma sorgente, tranne nel fatto che ogni riga è preceduta da un numero, che è il valore del contatore di programma della prima istruzione generata da quella riga.
Il messaggio di errore di esecuzione è accompagnato da questo numero ed è possibile fare un'approssimazione di dove si era quando l'errore è successo. L'unico problema con questo approccio è che, non appena si corregge un errore, il contatore di programma cambia e bisogna rielencare l'output del compilatore per avere i nuovi valori. Un altro compilatore ha la capacità di riportare il numero di linea, ma dovete aiutarlo, usando un'opzione, quando compilate il programma. Il punto è che dovete leggere il manuale fornito con il vostro compilatore, per vedere se è dotato di un dispositivo per aiutare la correzione.

Non lasciatevi sfuggire l'ovvio

È ora di dare un esempio. Quando facevo l'università mi guadagnavo un po' di soldi aggiustando dei televisori per i vicini e per la gente che aveva sentito dire che lo facevo. Una sera andai in una casa e vi trovai un televisore che non funzionava assolutamente. Controllai l'uscita di corrente alla presa e non ne trovai. Poi chiesi alla persona se aveva fatto fare dei lavori elettrici in casa recentemente. La risposta fu affermativa. Gli dissi che non c'era corrente a quella presa e provammo il televisore in un'altra. I tubi si accesero, ma non c'era alcuna immagine. Passai mezz'ora guardando diverse cose dentro il televisore e finalmente decisi che avrei dovuto portarlo in laboratorio e tracciare i segnali.

A casa, la sera seguente, trovai che mancava un segnale verticale.
Improvvisamente la spiegazione fu ovvia. Due tubi, i cui numeri di tipo non si leggevano più, erano stati scambiati. Ovviamente, queste persone avevano tolto i tubi dal televisore e li avevano portati al negozio vicino, dove c'era una macchina per controllarli. Quando li rimontarono, scambiarono quei due che sembravano uguali, ma non lo erano.
Un paio di anni di lavoro con i televisori mi avevano insegnato com'erano inseriti quei tubi e constatai che erano stati scambiati.

Li rimisi nelle posizioni giuste e il televisore funzionò. Lo ridiedi al cliente e gli dissi che se mi avesse informato di aver fatto controllare i tubi, avrei per prima cosa controllato se ce n'erano di scambiati e avrei risparmiato a tutti e due del tempo e delle spese. Il punto è che il proprietario del televisore non aveva pensato alla possibilità, ovvia, della mancanza di corrente, e poi aveva complicato le cose perché non mi aveva dato abbastanza informazioni, ovvie e sufficienti, tali da permettermi di trovare subito una soluzione al problema.

Quando correggete un programma, non lasciatevi sfuggire gli errori ovvii, come una variabile non inizializzata. A volte ho trovato di non aver chiamato una procedura in qualche punto nel programma.
Un punto e virgola collocati male possono anche causare dei risultati strani. Se dichiarate una variabile (in una procedura), che ha un nome identico a una variabile globale, e poi cercate di accedere a quella globale dalla procedura, possono succedere cose assolutamente inusuali all'interno del vostro programma, poiché la variabile locale ha la precedenza e quindi blocca l'accesso a quella globale. Il risultato è un problema che assomiglia a una variabile non inizializzata. Se cercate di far stampare il valore della variabile all'interno della procedura, potreste capire la causa della difficoltà. Questo tipo di errore è particolarmente facile da correggere, se avete usato il BASIC per molto tempo e siete abituati ad avere solo variabili globali, senza preoccuparvi di precedenze.

Alcuni sistemi Pascal non vi permettono di dichiarare una costante, per esempio NUMERO = 64000;. Il TYPE di costante è deciso dal valore e questo sembra un INTEGER. Il numero, però, è troppo grande da poter essere rappresentato come un intero, e quindi c'è un errore. Bisogna usare NUMERO = 6.4E4;, l'equivalente di 64000 in notazione scientifica, se volete rappresentare questo numero come costante, per segnalare al compilatore che è un numero REAL, e non un INTEGER. ■

Sprite Editor per il TI99/4 A

PERSONAL SOFTWARE

Un programma veramente efficiente per definire i caratteri grafici

di *Filippo Cerulo*

Tutti i possessori di TI99/4 A si trovano, prima o poi, di fronte alla necessità di definire nuovi caratteri da utilizzare nei programmi. Il procedimento è relativamente semplice e senz'altro più accessibile al principiante di analoghe costruzioni possibili su altri home computer. Ciò non toglie che quando il numero di caratteri da ridefinire è elevato il calcolo "manuale" dell'identificatore di sagoma esadecimale sia molto noioso e stancante. Inoltre, in programmi che usano sprite a grandezza doppia ogni sagoma relativa è formata da quattro caratteri: quindi deve essere costruita su una griglia formata da 16x16 pixel ed identificata da una stringa esadecimale lunga ben 64 caratteri. A questo punto sembra inevitabile demandare questi gravosi (e tediosi) compiti al proprio computer: questo è esattamente ciò che ho fatto.

Qualche cenno sulla CALL CHAR

L'argomento ridefinizione di caratteri è stato già ampiamente trattato anche sulle pagine di questa rivista ed inoltre il manuale, almeno su questo punto, è abbastanza chiaro.

Nel TI99 tutti i caratteri il cui codice ASCII è compreso tra 32 e 159 possono essere ridefiniti, cioè può essere modificata la loro sagoma. Ad ogni sagoma è associato un *identificatore* formato da sedici cifre esadecimali. La sagoma di ogni carattere viene costruita su una griglia di 8x8 pixel. Ogni cifra esadecimale dell'identificatore rappresenta, secondo la tabella riportata nel manuale, 4 pixel in sequenza da destra a sinistra e dall'alto in basso. Con l'istruzione:
CALL CHAR (codice di carattere, "identificatore di sagoma")
si ridefinisce un carattere. Questa istruzione può essere usata anche come *comando* per i caratteri il cui codice sia compreso tra 128 e 159. Cioè, se si dà ad esempio il comando:
CALL CHAR (129, "3C4299A-1A199423C")
il simbolo di copyright diventa disponibile sulla tastiera premendo CTRL A. In ogni caso, i caratteri il cui codice sia compreso tra 32 e 127 possono essere ridefiniti da programma, ma quando si torna in "modo comando" (cioè ad ogni stop del programma) riprendono le sagome originali.

Le differenze del BASIC esteso

In Extended BASIC possono essere ridefiniti solo i caratteri il cui codice ASCII sia compreso tra 32 e 143. Inoltre l'istruzione CALL CHAR, oltre il modo usuale, ha altre due costruzioni sintattiche. Le illustreremo con due esempi. L'istruzione:
100 CALL CHAR
(65, "080C0EFFFF0E0C083C4299
A1A199423C")
ridefinisce i caratteri "A" e "B" come, rispettivamente, una freccia ed il simbolo di copyright. In BASIC esteso, infatti, l'identificatore può essere lungo fino a 64 cifre esadecimali e ridefinisce il carattere il cui codice è specificato nella istruzione, nonchè i caratteri seguenti, usando sedici cifre esadecimali per ogni carattere. L'istruzione:
CALL CHAR (65, "080C0EFFF-F0E080C", 78, "000000FFFF")
ridefinisce i caratteri "A" ed "N". Possiamo cioè ripetere codici ed identificatori all'interno delle parentesi senza riscrivere la CALL CHAR. Notate, nel caso del codice 78, che gli zeri finali di un identificatore esadecimale possono essere omessi. Un'altra caratteristica interessante (comune anche ad altri BASIC) è la funzione RPT$. Si tratta di una funzione di stringa con la seguente sintassi:
RPT$ ("stringa", numero di ripetizioni)
Ad esempio l'istruzione:
200 A$ = RPT$ ("0C",8)
equivale a:
200 A$ = "0C0C0C0C0C0C0C0C"
In pratica, essa assegna ad una variabile di stringa una ripetizione di caratteri determinati. Può essere usata (come tutte le funzioni di stringa) direttamente in una CALL CHAR, come nell'esempio seguente:
200 CALL CHAR(129, RPT$("F", 16))
Questa istruzione ridefinisce il carattere 129, "accendendo" tutti i pixel che lo compongono. Un cenno va fatto anche ad un'altra istruzione molto utile. La sua sintassi è:
CALL CHARPAT (codice di carattere, variabile di stringa)
Essa assegna a "variabile di stringa" l'identificatore esadecimale della sagoma rappresentante il carattere il cui codice ASCII compare nell'istruzione. Ad esempio:
200 CALL CHARPAT (65,A$)
assegna ad A$ il valore esadecimale

Sprite Editor per il TI99/4A

Listato 1. *Il programma BASIC.*

```
100 !**** SPRITE EDITOR ****
110 !
120 !(C) 1984 - FILIPPO CERULO - ARCADE
130 !
140 !TEL. 0824/871190
150 !
160 DIM BN$(15),HE$(15)
170 FOR I=0 TO 15 :: READ BN$(I),HE$(I):: NEXT I
180 DATA 0000,0,0001,1,0010,2,0011,3,0100,4,0101,5,0110,6,0111,7
190 DATA 1000,8,1001,9,1010,A,1011,B,1100,C,1101,D,1110,E,1111,F
200 FOR I=104 TO 109 :: READ A$ :: CALL CHAR(I,A$):: NEXT I :: CALL COLOR(10,6,1
)
210 DATA 000000FFFF,1818181818181818,000000F8F8181818,181818F8F8,0000001F1F18181
8,1818181F1F
220 CALL CLEAR :: CALL CHAR(91,"3C4299A1A199423C",93,"000000FF",132,"00003C3C3C3
C",139,"FF818181818181FF",140,RPT$("F",16))
230 CALL COLOR(14,3,1):: CALL SCREEN(15)
240 GOSUB 770 :: CALL CHARPAT(65,B$,66,C$):: CALL SCRIVI(7,3,B$,BN$()):: CALL SC
RIVI(15,3,C$,BN$()):: CALL STAMPA(8):: CALL GET(COD)
250 CALL CLEAR :: GOSUB 770 :: CALL STAMPA(2):: CALL MASCHERA
260 DATA 3,8,"SPRITE  EDITOR",4,8,]]]]]]]]]]]]]],8,14,EXTENDED BASIC,10,14,SOFTW
ARE,13,14,[ 1984 BY
270 DATA 15,14,ARCADE,19,14,PRESS ANY,21,14,KEY,1,7," SPRITE  EDITOR hhhhhh",2,8
,]]]]]]]]]]]]]]
280 CALL CANCELLA :: RESTORE 290 :: CALL STAMPA(6)
290 DATA 6,19,SCEGLI :,9,19,1 WRITE,11,19,"2 DISPLAY",13,19,3 STRINGA,15,19,"4 N
EW",17,19,5 FINE
300 CALL GET(COD):: IF COD<49 OR COD>53 THEN 300
310 ON COD-48 GOSUB 350,490,560,660,330
320 GOTO 280
330 CALL CLEAR :: STOP ! FINE LOGICA DEL PROGRAMMA
340 REM **** SBR WRITE ****
350 CALL CANCELLA :: RESTORE 360 :: CALL STAMPA(5)
360 DATA 6,19,WRITE,9,20,1 OVER,11,20,2 SET,13,20,3 RESET,15,20,4 STOP
370 CALL MAGNIFY(1):: CALL SPRITE(#2,132,5,33,17):: X=33 :: Y=17 :: CAR=0
380 CALL LOCATE(#2,X,Y)
390 CALL KEY(5,COD,ST):: IF ST=0 THEN 390 ELSE IF POS("1234ESDXWRZC",CHR$(COD),1
)=0 THEN 390
400 IF COD=52 THEN CALL DELSPRITE(#2):: RETURN
410 IF COD>52 THEN 450
420 IF COD=49 THEN CAR=0 :: CALL COLOR(#2,5):: GOTO 390
430 IF COD=50 THEN CAR=140 :: CALL COLOR(#2,2):: GOTO 390
440 IF COD=51 THEN CAR=139 :: CALL COLOR(#2,16):: GOTO 390
450 CALL POSITION(#2,X,Y):: IF CAR<>0 THEN CALL HCHAR(INT(X/8)+1,INT(Y/8)+1,CAR)
460 C$=CHR$(COD):: X=X+8*(-(POS("WER",C$,1)<>0)*(X>40)+(POS("ZXC",C$,1)<>0)*(X<1
46))
470 Y=Y+8*(-(POS("SWZ",C$,1)<>0)*(Y>24)+(POS("DRC",C$,1)<>0)*(Y<130)):: GOTO 380
480 REM **** SBR DISPLAY ****
490 CALL CANCELLA :: DISPLAY AT(3,20):"DISPLAY" :: DISPLAY AT(23,1)SIZE(18)BEEP:
"ATTENDERE PREGO !"
500 RESTORE 510 :: PAT$=""
510 DATA 5,3,5,19,13,3,9,19,5,11,13,19,13,11,17,19
520 FOR I=1 TO 4 :: READ R,C,R1,C1 :: CALL LEGGI(R,C,ID$,HE$()):: PAT$=PAT$&ID$
530 DISPLAY AT(R1,C1):SEG$(ID$,1,8):: DISPLAY AT(R1+2,C1)BEEP:SEG$(ID$,9,8):: NE
XT I
```

Sprite Editor per il TI99/4A

Seguito listato 1.

```
540 GOSUB 680 :: RETURN
550 REM **** SBR STRINGA ****
560 CALL CANCELLA :: CALL MASCHERA :: CALL CHAR(128,RPT$("0",64)):: DISPLAY AT(3
,22):"STRINGA"
570 COD=128 :: PAT$="" :: RESTORE 580
580 DATA 5,3,13,3,5,11,13,11
590 FOR RI=5 TO 19 STEP 4 :: CALL HCHAR(23,2,32,18):: CALL CHARPAT(COD,B$):: DIS
PLAY AT(RI,19):SEG$(B$,1,8)
600 DISPLAY AT(RI+2,19):SEG$(B$,9,8):: ACCEPT AT(RI,19)SIZE(-8)VALIDATE(DIGIT,"A
BCDEF")BEEP:B1$
610 ACCEPT AT(RI+2,19)SIZE(-8)VALIDATE(DIGIT,"ABCDEF")BEEP:B2$ :: B$=B1$&B2$ ::
PAT$=PAT$&B$
620 READ R,C :: IF B$=RPT$("0",16)THEN 640
630 DISPLAY AT(23,1)BEEP:"ATTENDERE PREGO !" :: CALL SCRIVI(R,C,B$,BN$())
640 COD=COD+1 :: NEXT RI :: GOSUB 680 :: RETURN
650 REM **** SBR NEW ****
660 CALL CANCELLA :: DISPLAY AT(23,4)BEEP:"NEW" :: CALL MASCHERA :: CALL HCHAR(2
3,4,32,15):: RETURN
670 REM *** MOSTRA GLI SPRITES ***
680 CALL CHAR(128,PAT$):: CALL HCHAR(23,3,32,28)
690 DISPLAY AT(23,1)SIZE(8):"COLORE ?" :: ACCEPT AT(23,11)SIZE(2)VALIDATE(DIGIT)
BEEP:COL
700 IF COL>16 THEN 690 ELSE IF COL=0 THEN CALL DELSPRITE(#1):: RETURN
710 CALL HCHAR(22,20,32,10):: CALL HCHAR(23,20,32,10):: CALL DELSPRITE(#1)
720 CALL COLOR(13,COL,1):: DISPLAY AT(22,25):"";"" :: DISPLAY AT(23,25):"";"
"
730 CALL SPRITE(#1,128,COL,1,161,10,0):: CALL MAGNIFY(4)
740 CALL COINC(#1,161,161,1,A):: IF A=0 THEN 740
750 CALL MOTION(#1,0,0):: CALL LOCATE(#1,161,161):: GOTO 690
760 REM **** STAMPA CORNICE ****
770 CALL HCHAR(24,1,104,64):: CALL VCHAR(1,32,105,48):: CALL HCHAR(1,1,108):: CA
LL HCHAR(1,32,106)
780 CALL HCHAR(24,1,109):: CALL HCHAR(24,32,107):: RETURN
790 REM **** SUB STAMPA ****
800 SUB STAMPA(NU):: FOR I=1 TO NU :: READ R,C,S1$ :: DISPLAY AT(R,C):S1$ :: NEX
T I :: SUBEND
810 REM **** SUB GET ****
820 SUB GET(COD):: CALL SOUND(100,1047,6)
830 CALL KEY(5,COD,ST):: IF ST=0 THEN 830
840 SUBEND
850 REM **** SUB MASCHERA ****
860 SUB MASCHERA :: FOR R=5 TO 20 :: CALL HCHAR(R,3,139,16):: NEXT R :: SUBEND
870 REM **** SUB CANCELLA ****
880 SUB CANCELLA :: FOR R=3 TO 23 :: CALL HCHAR(R,20,32,12):: NEXT R :: CALL HCH
AR(23,2,32,30):: CALL HCHAR(22,2,32,30):: SUBEND
890 REM **** SUB BIN ****
900 SUB BIN(A$,N):: N=0 :: FOR I=3 TO 0 STEP -1 :: N=N+VAL(SEG$(A$,4-I,1))*2^I :
: NEXT I :: SUBEND
910 REM **** SUB LEGGI ****
920 SUB LEGGI(R,C,A1$,B1$()):: A1$="" :: FOR R1=R TO R+7 :: FOR C1=C TO C+7 STEP
 4 :: A$="" :: FOR C2=C1 TO C1+3
930 CALL GCHAR(R1,C2,COD):: A$=A$&CHR$(COD-91):: NEXT C2
940 CALL BIN(A$,N):: A1$=A1$&B1$(N):: NEXT C1 :: NEXT R1 :: SUBEND
950 REM **** SUB ESA ****
960 SUB ESA(R,C,A$):: FOR I=1 TO 4 :: CALL HCHAR(R,C-1+I,ASC(SEG$(A$,I,1))+91)::
 NEXT I :: SUBEND
970 REM **** SUB SCRIVI ****
980 SUB SCRIVI(R,C,A$,A1$()):: A=1 :: FOR R1=R TO R+7 :: FOR C1=C TO C+7 STEP 4
:: B$=SEG$(A$,A,1)
990 IF ASC(B$)<58 THEN K=VAL(B$)ELSE K=ASC(B$)-55
1000 CALL ESA(R1,C1,A1$(K)):: A=A+1 :: NEXT C1 :: NEXT R1 :: SUBEND
```

Sprite Editor per il TI99/4A

"003844447C444444". Infine l'istruzione:

200 CALL CHARSET

usata in un programma ricarica le sagome normali in caratteri il cui codice sia compreso tra 32 e 127 e che siano stati precedentemente modificati.

Costruire uno sprite

Il programma che vi presento ha come scopo principale quello di calcolare l'identificatore esadecimale relativo alle sagome (esattamente quattro) che potete costruire, attraverso facili procedure, su una griglia di 16x16 pixel ingrandita otto volte. Alla fine il computer vi mostra lo sprite (o il carattere) creato in movimento e con il colore voluto, sia in grandezza normale che doppia. Potete inoltre far costruire sulla griglia il carattere relativo ad un identificatore da voi inserito (anche eventualmente per modificarlo). Vediamo un po' come operare.

Dopo la sigla il computer stampa la griglia ed il menu principale. Le scelte sono accettate tramite istruzione CALL KEY e non INPUT, per evitarvi di premere ENTER ogni volta: quindi non vi aspettate un cursore lampeggiante, ma premete semplicemente il tasto numerico relativo. Esaminiamo ora, una per una, le opzioni possibili.

1 WRITE: vi permette di costruire sulla griglia la sagoma desiderata. La "guida" è uno sprite, a forma di quadratino, che potete muovere utilizzando in modo usuale i tasti W, E, R, S, D, Z, X, C. La guida è sufficientemente veloce e può muoversi anche in diagonale. In modo WRITE (scrivi) sono possibili quattro opzioni e precisamente: OVER (1): il quadratino è blu e si muove sulla griglia senza altre conseguenze; SET (2): il quadratino è nero e "accende" (setta) i pixel al suo passaggio; RESET (3): il quadratino è bianco e "spegne" al suo passaggio i pixel eventualmente accesi; FINE (4): ritorna al menu principale.

Per selezionare l'opzione desiderata basta premere il tasto numerico relativo. Con un po' di pratica la fase di "costruzione" della sagoma diventa abbastanza semplice e perfino divertente. Potete creare una sagoma formata da quattro caratteri (per uno sprite "double sized") o anche quattro sagome diverse, ognuna di un carattere.

2 DISPLAY: calcola l'identificatore esadecimale della sagoma presente sulla griglia. Le cifre esadecimali sono mostrate a gruppi di otto, sulla destra dello schermo, e a ogni due gruppi corrisponde un carattere secondo lo schema:

1 3 cioè dall'alto in basso e da destra a sinistra. Quindi il
2 4 computer vi chiede il colore con il quale visualizzare la sagoma. Dovrete perciò introdurre il codice del colore desiderato
(un intero tra uno e sedici) ed il computer vi mostrerà in basso a destra la sagoma in grandezza normale, mentre uno sprite in grandezza doppia si muoverà in verticale e si fermerà alla base dello schermo. A questo punto potrete cambiare ancora il colore della sagoma o tornare al menu principale, introducendo zero come codice di colore. In ogni caso la sagoma sulla griglia non sarà modificata.

3 STRINGA: questa opzione vi permette di visualizzare sulla griglia la sagoma relativa ad un identificatore da voi inserito. Ogni eventuale sagoma preesistente viene cancellata e dovete introdurre il vostro identificatore sulla destra dello schermo a gruppi di otto cifre. Ogni sedici cifre viene visualizzato un carattere secondo la solita corrispondenza. Alla fine potrete vedere la vostra sagoma in grandezza normale o doppia, secondo le modalità dell'opzione DISPLAY.

4 NEW: cancella la sagoma eventualmente esistente sulla griglia.

5 FINE: pone termine al programma.

Dopo l'esecuzione di uno qualsiasi dei sottoprogrammi DISPLAY e STRINGA potete tornare in modo WRITE per correggere la sagoma e ricalcolarne l'identificatore. Infatti, in generale, la prima volta che si costruisce una sagoma, essa non sarà esente da imperfezioni. Inoltre, la scelta del colore è fondamentale nell'esaltare o meno alcuni particolari. Considerando anche che il colore di sfondo durante il programma è grigio, vi consiglio di evitare, quando valutate la sagoma, il nero e di orientarvi su colori leggermente più sfumati (blu o verde scuro).

Quattro passi nel listato

Le caratteristiche peculiari dell'Extended BASIC, rispetto ai linguaggi implementati su altri home computer, sono molte. Oltre alla grafica ed agli sprite, infatti, disponiamo di una grande versatilità nelle istruzioni di INPUT/OUTPUT su video e soprattutto abbiamo la possibilità di scrivere sottoprogrammi in BASIC indipendenti dal programma principale e sostanzialmente diversi dalle normali subroutine. Questi sottoprogrammi sono attivati da istruzioni del tipo:
200 CALL nome di sottoprogramma (lista di parametri)
Lo spazio non mi permette di dilungarmi oltre su questi aspetti, peraltro interessantissimi, del BASIC esteso e rimando gli approfondimenti ad altre occasioni. Basti sapere che le variabili usate nei sottoprogrammi sono "esclusive", nel senso che non hanno nulla a che vedere con eventuali altre variabili con lo stesso nome presenti nel programma principale.

Cominciamo appunto ad esaminare i sottoprogrammi usati. Vi accorgerete che alcuni sono di utilizzo generale e possono essere trasportati in altri programmi, esattamente come sono scritti.
SUB STAMPA(NU) - Serve a stampare una serie di stringhe (esatta-

Sprite Editor per il TI99/4A

mente NU), inserite nel programma con istruzioni DATA, con la sintassi:
DATA riga, colonna, stringa (,....)
Risparmia molto spazio e permette una facile impostazione di maschere video o di menu anche molto complessi.
SUB GET (COD) - Simula l'istruzione GET (o INKEY$) presente su altri computer. Il programma si ferma, emette un suono e aspetta che premiate un qualsiasi tasto. Il codice ASCII del tasto premuto è ritornato nella variabile COD.
SUB MASCHERA - Stampa la griglia su cui costruire la sagoma.
SUB CANCELLA - Serve a cancellare la parte destra ed il bordo inferiore dello schermo.
SUB BIN(A$,N) - Converte un numero binario di quattro cifre, contenuto nella variabile di stringa A$, nel corrispondente decimale N. È un sottoprogramma di uso universale.
SUB LEGGI (R,C,A1$, B1$()) - Calcola l'identificatore di sagoma del carattere presente su una parte (8x8 pixel) della griglia. Accetta in input le coordinate (R,C), da cui cominciare il calcolo, ed il vettore B1$(I), che corrisponde a HE$(I) del programma principale. Ritorna in A$ l'identificatore. Legge i caratteri sul video con una serie di CALL GCHAR e costruisce ogni volta un numero binario di quattro cifre. Quindi converte il numero binario nel decimale N: il valore esadecimale corrispondente è memorizzato nella variabile B1$(N).
SUB ESA (R,C,A$) - Visualizza sulla griglia i quattro pixel relativi al numero binario contenuto nella variabile A$ indicando la riga e la colonna da cui cominciare.
SUB SCRIVI (R,C,A$,A1$()) - Visualizza sulla griglia il carattere corrispondente all'identificatore contenuto nella variabile A$ indicando la riga e la colonna da cui cominciare. Il vettore A1$(I) corrisponde a BN$(I) del programma principale.

Diamo ora un'occhiata al listato

Le righe 170 - 190 caricano i contenuti dei vettori BN$(I) e HE$(I). Le righe 200 - 230 ridefiniscono i caratteri necessari e determinano i colori dei caratteri e dello sfondo.
La sigla è compresa nella riga 240: i caratteri che in essa appaiono ingranditi sono "A" e "B" (codici 65 e 66), ma ognuno di voi può personalizzare il programma, modificando i codici ASCII nell'istruzione CALL CHARPAT.
La riga 250 stampa la griglia e l'intestazione e la 280 il menu principale.

Dopo la scelta, accettata alla linea 300, il programma passa il controllo alla subroutine desiderata. In particolare:
la subroutine WRITE (350 - 470) crea la "guida" (370) e la muove sulla griglia. Si noti che le istruzioni che muovono "fisicamente" lo sprite, a seconda del tasto premuto, sono soltanto due, la 460 e la 470 e ciò, oltre a risparmiare spazio, velocizza di molto il programma. Le altre istruzioni, invece, provocano l'effetto desiderato (accensione o spegnimento di pixel), a seconda del modo scelto.
La subroutine DISPLAY (490 - 540) calcola, con quattro chiamate del sottoprogramma LEGGI, l'identificatore della sagoma presente sulla griglia.
La subroutine STRINGA (560 - 640) accetta, sulla destra dello schermo, l'identificatore che inserite e visualizza la sagoma relativa, chiamando il sottoprogramma SCRIVI.
La subroutine NEW (660) cancella la sagoma esistente sulla griglia. Inoltre sono più volte utilizzate, nel corso del programma, le subroutine comprese tra le seguenti righe:
680 - 750, che mostra i caratteri costruiti sulla griglia nel colore desiderato. I codici dei caratteri usati sono 128, 129; 130,131. Nella riga 720, tra le virgolette, vanno inseriti i tasti di controllo relativi ai suddetti codici, e quindi le due istruzioni che mostrano il carattere a grandezza normale vanno scritte come segue:
DISPLAY AT(22,25): "(CTRL ,)"; "(CTRL B)"
DISPLAY AT(23,25): "(CTRL A)";"(CTRL C)"
770 - 780, che stampa la "cornice dello schermo".

E per finire....

Il programma non è molto lungo (circa 5 Kbyte), ma si rivelerà sicuramente utile a quanti provvederanno a digitarlo. Ho fatto particolare attenzione a costruire il listato nel modo più strutturato possibile, anche perché il BASIC esteso facilita l'uso di simili tecniche. Quindi ogni modifica che vorrete eventualmente apportare dovrebbe risultare agevole. La strada migliore per impadronirsi di un linguaggio, infatti, è quella di usarlo il più possibile, sia scrivendo propri programmi che modificando, secondo le proprie esigenze, quelli degli altri e l'Extended BASIC non sfugge di certo a questa regola. Buon lavoro! ■



Libri firmati JACKSON



66 PROGRAMMI PER ZX81 e ZX80 CON NUOVA ROM + HARDWARE
Come sfruttare tutte le capacità degli ZX e, addirittura, moltiplicarle
144 pag. **L. 12.000**
Cod. **520D**

PROGRAMMI DI MATEMATICA E STATISTICA
Come acquistare la logica necessaria a risolvere con metodo, senza perdite di tempo, i problemi con il calcolatore
228 pag. **L. 16.000**
Cod. **552D**

SINFONIA PER UN COMPUTER VIC 20
Imparare divertendosi è la prerogativa di questo libro: prima giocate e poi date un'occhiata ai listati dei programmi
122 pag. **L. 10.000**
Cod. **563D**

75 PROGRAMMI IN BASIC PER IL VOSTRO COMPUTER
Programmi sperimentati e pronti da usare, oppure da rielaborare, ampliare, modificare, assemblare
196 pag. **L. 12.000**
Cod. **551D**

77 PROGRAMMI PER SPECTRUM
Dalla Grafica alla Business Grafica, dalla musica alle animazioni, dai giochi all'Elettronica ... tutte le possibilità offerte dallo Spectrum
150 pag. **L. 16.000**
Cod. **555A**

GIOCHI GIOCHI GIOCHI PER IL VOSTRO VIC 20
Il modo più divertente di avvicinarsi al computer. Una trentina di giochi per il vostro VIC 20
108 pag. **L. 9.000**
Cod. **557D**

PROGRAMMI UTILI PER IBM PC
Oltre 65 programmi ampiamente collaudati e pronti per l'uso per il Personal Computer IBM
176 pag. **L. 15.000**
Cod. **564D**

La Biblioteca che fa testo

GRUPPO EDITORIALE JACKSON

Attenzione compilare per intero la cedola
ritagliare (o fotocopiare) e spedire in busta chiusa a:
GRUPPO EDITORIALE JACKSON
Divisione Libri
Via Rosellini, 12 - 20124 Milano

CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale
		Totale	

☐ Pagherò contrassegno al postino il prezzo indicato più **L. 2000 per contributo fisso spese di spedizione**

Condizioni di pagamento con esenzione del contributo spese di spedizione:

☐ Allego assegno della Banca ______________________

n° ______________________

☐ Allego fotocopia del versamento su c/c n. 11666203 a voi intestato

☐ Allego fotocopia di versamento su vaglia postale a voi intestato

Nome ______________________
Cognome ______________________
Via ______________________
Cap ______ Città ______________ Prov. ______
Data ______ Firma ______________________

Spazio riservato alle Aziende. Si richiede l'emissione di fattura

Partita I.V.A. |_|_|_|_|_|_|_|_|_|_|_|

ORDINE MINIMO L. 50.000

Uso delle tabelle con TI99/4A

Sono di uso molto frequente nei programmi ad orientamento commerciale

di *Sergio Borsani*

Introduzione

Contro l'opinione dominante che vede in un home computer uno strumento per il tempo libero ed il gioco, le domande che più di frequente mi sono state rivolte riguardano soprattutto applicazioni di tipo commerciale oltre che di natura didattico-scientifica. Molte persone conducono attività che per le limitate dimensioni non giustificano l'acquisto di un personal da sette o otto milioni, oppure, prima di una spesa impegnativa, vogliono capire quali possono essere le prestazioni di un computer e quale la sua attività. In pratica vogliono sperimentare su piccola scala alcune applicazioni per acquisire gli elementi che potranno giustificare la scelta futura. Il programma del listato 1 si inserisce nell'area della gestione di magazzino ed è soltanto una parte semplificata di un programma completo che assolve il solo compito di riepilogare gli articoli venduti e determinarne il valore parziale e totale.

Con ogni probabilità l'argomento interessa non solo i possessori del TI99/4A ma chiunque abbia un piccolo computer. Ebbene, ho usato il TI BASIC invece dell'Extended BASIC proprio per rendere più semplice la sua comprensione e più facile la sua conversione in un'altra versione di BASIC. Nella figura 6 vengono indicate le modifiche necessarie per rendere il programma compatibile con i computer VIC 20 e Commodore 64. La memoria minima utilizzata non supera i 3 Kbyte. Ben più importante a questi fini è lo spazio occupato dalla tabella. Il TI99/4A, senza espansione di memoria può contenere una tabella di 500 x 2 elementi, cioè, ad esempio, 500 numeri di codice e i rispettivi 500 prezzi unitari; con l'espansione di memoria non si riesce, come speravo, a raddoppiare tale numero ma si arriva a gestire una tabella di 900 x 2 elementi anche se rimane ancora molta memoria disponibile per il programma.

Che cos'è una tabella

Un semplice diagramma di flusso può rappresentare un file-archivio, un file-dati e l'elaborazione di un file-archivio aggiornato. Un altro schema altrettanto elementare prevede un file-dati e l'elaborazione di un riepilogo con la consultazione di una tabella. Questa è una struttura formata da variabili con indice e viene altrimenti chiamata matrice, o vettore, quando la matrice sia ad una dimensione. Non va quindi confusa con i "tabelloni elettronici" (spreadsheet) dei programmi come il VisiCalc o il Multiplan. L'uso delle variabili con indice e delle matrici costituisce una svolta significativa per quanti coltivano l'hobby della programmazione ed anche per chi alla programmazione si accosta per motivi professionali. Generalmente accade che dopo aver imparato le istruzioni più comuni di assegnazione, scrittura, salto condizionato e incondizionato, il primo vero scoglio nel quale ci si imbatte è costituito dalle matrici; una volta superato, tuttavia, i programmi acquistano una maggiore complessità e diventano più potenti.

Una tabella va intesa come un file residente in memoria centrale i cui elementi vengono consultati durante l'elaborazione di un altro file residente su nastro. Ovviamente potrebbe risiedere su un dischetto ma, se è possibile, preferisco riferire gli esempi alla configurazione di sistema più diffusa, cioè: console, televisore e registratore a cassette. In questo programma la tabella da consultare contiene i codici di un certo numero di articoli ed i relativi prezzi unitari. Poiché sarebbe assurdo dover scrivere questi dati ogni volta



Uso delle tabelle con TI99/4A

che si avvia il programma, anche la tabella va memorizzata una volta per tutte su nastro e richiamata prima di ogni elaborazione per la quale sia richiesta. Quando la tabella è stata caricata nella memoria centrale si può eseguire il programma principale il quale può valorizzare un certo numero di articoli movimentati ricavando la quantità da un file sequenziale e desumendo il prezzo unitario dalla tabella precedentemente caricata. In questo caso specifico l'output è dato sul video, ma chi possiede una stampante può, con l'aggiunta di poche istruzioni, ottenere una copia permanente dell'elaborato su carta. Chi utilizza contemporaneamente due registratori, può utilizzarne uno in lettura ed uno per salvare i risultati dell'elaborazione. Inoltre chi volesse utilizzare il programma con un numero di record elevato dovrà introdurre un'istruzione CALL KEY per interrompere lo scrolling verso l'alto e per proseguire dopo la lettura di una "videata" senza dover ogni volta interrompere il programma con il comando FCTN (4) e farlo riprendere con il comando CON (ENTER).

Come funziona il programma

Il programma si apre con il menu riportato nella figura 1. I punti 1, 2, 3 e 4 servono per inizializzare i file che verranno utilizzati dal programma principale e per controllare che il loro contenuto sia corretto. L'elaborazione avviene al punto 5 (STAMPA RIEPILOGO) e prevede necessariamente l'esistenza dei due file (file Tabella e file Movimenti) creati selezionando da menu i punti 1 e 3. Si scelga quindi il punto 1. Compariranno numeri progressivi e la richiesta di numero di codice e prezzo unitario. Ci si limiti a numeri

```
1. CREA TABELLA
2. STAMPA TABELLA
3. CREA FILE MOVIMENTI
4. STAMPA FILE MOVIMENTI
5. STAMPA RIEPILOGO
6. FINE PROGRAMMA
```

Figura 1. *Il menu in apertura di programma. È necessaria l'inizializzazione del file Tabella e del file Movimenti per l'elaborazione del riepilogo.*

```
TABELLA

COD.: 1001        PREZZO  12000
COD.: 1002        PREZZO    750
COD.: 1003        PREZZO   1500
COD.: 1004        PREZZO   8400
COD.: 1005        PREZZO   6000

 premi space per continuare
```

Figura 2. *Selezionando il tasto 2 si ottiene la stampa del contenuto della tabella, nel caso specifico i numeri di codice degli articoli ed i prezzi unitari. Non è necessario che i codici siano progressivi ma devono essere ordinati in ordine crescente. In configurazione base il TI99/4A può gestire 500 articoli.*

di codice con non più di quattro o cinque caratteri altrimenti in seguito sorgeranno problemi di tabulazione sul limitato display a 28 colonne del nostro computer. Ed ancora, la matrice che costituirà la tabella è stata dimensionata con un massimo di 100 elementi; l'introduzione di un numero maggiore a 100 elementi di tabella provocherebbe in seguito la comparsa del messaggio d'errore BAD SUBSCRIPT per il riferimento ad un indice superiore a quello ammesso dal dimensionamento. A titolo esemplificativo si introducano i dati riportati nella figura 2. I numeri di codice vanno dal 1001 al 1005 ed i rispettivi prezzi unitari sono

```
FILE MOVIMENTI
CODICE: 1002   QUANTITA' 34
CODICE: 1002   QUANTITA'  6
CODICE: 1002   QUANTITA'  3
CODICE: 1004   QUANTITA' 12
CODICE: 1004   QUANTITA'  6
CODICE: 1005   QUANTITA'  3
CODICE: 1005   QUANTITA' 10
CODICE: 1005   QUANTITA'  5
CODICE: 1005   QUANTITA'  6
 premi space per continuare
```

Figura 3. *Display nella fase di controllo del file Movimenti.*
Più record possono riguardare lo stesso articolo e, mancando una routine di ordinamento, si avrà l'attenzione di scrivere i numeri di codice in ordine crescente.

```
         RIEPILOGO

CODICE   QTA   PREZZO    VALORE
-------------------------------
 1002     43     750      32250
 1004     18    8400     151200
 1005     24    6000     144000
-------------------------------
              TOTALE     327450
-------------------------------

 premi space per continuare
```

Figura 4. *Come appare l'elaborato finale. Tutti i movimenti riguardanti lo stesso articolo vengono presentati sotto un'unica voce comprendente il numero di codice dell'articolo, la quantità (QTA), il prezzo unitario ed il valore complessivo.*
Il prezzo unitario viene desunto dal programma stesso con la consultazione di una tabella caricata in memoria nella fase iniziale.

12000, 750, ecc. Per indicare al computer che si è terminato bisogna scrivere FINE alla successiva richiesta di codice. Naturalmente, durante questa prima fase di inizializzazione della tabella, è necessario seguire le indicazioni che appaiono sul video

per l'uso del registratore. È altresì importante segnare le posizioni indicate dal contanastro (tape counter) per poter ritrovare inequivocabilmente il file appena registrato e non sovrapporre a questo altre registrazioni. Ritornati al menu si potrà controllare il contenuto del file Tabella selezionando il numero 2. Sul video si vedrà quanto è illustrato dalla figura 2. Premendo poi la barra spaziatrice (o qualsiasi altro tasto) si ritorna al menu principale per passare alla fase successiva: la creazione del file Movimenti. Questo potrebbe rappresentare gli articoli venduti nel corso della giornata, ogni suo record contiene un numero di codice e la quantità. È anche naturale presupporre che lo stesso articolo sia stato venduto più volte e che pertanto nel file Movimenti più record contengano lo stesso numero di codice. Inoltre, non tutti gli articoli della tabella verranno necessariamente movimentati, ma tutti gli articoli movimentati dovranno avere i codici presenti in tabella.

Importante ai fini del corretto funzionamento del programma è che i numeri di codice siano in ordine crescente! Se si desidera creare un file Movimenti con i codici non ordinati sarà necessario aggiungere una routine per l'ordinamento in modo da operare sempre con file ordinati per numero crescente di codice. Per esempio, i dati potrebbero essere desunti dalla figura 3.

Mentre il file Tabella viene caricato in memoria centrale e deve sottostare ai limiti imposti dal dimensionamento, il file Movimenti è condizionato solo dalla lunghezza del nastro magnetico in quanto i suoi record vengono letti ed elaborati uno alla volta. La figura 3 riporta il display durante la fase di controllo e stampa del file Movimenti (opzione 3 del menu). Anche in questo caso non si dimentichi di segnare i valori indicati dal contanastro. L'elaborazione consiste nel caricare in memoria la tabella, leggere ad uno ad uno gli articoli movimentati, sommare le quantità che si riferiscono ad uno stesso articolo, ricavare dalla tabella il prezzo unitario, calcolare il valore (quantità x prezzo unitario) per ogni articolo ed infine stampare il valore totale. L'utente, selezionato il punto 5 del menu, dovrà solo seguire le indicazioni relative all'uso del registratore, il riepilogo apparirà come nella figura 4.

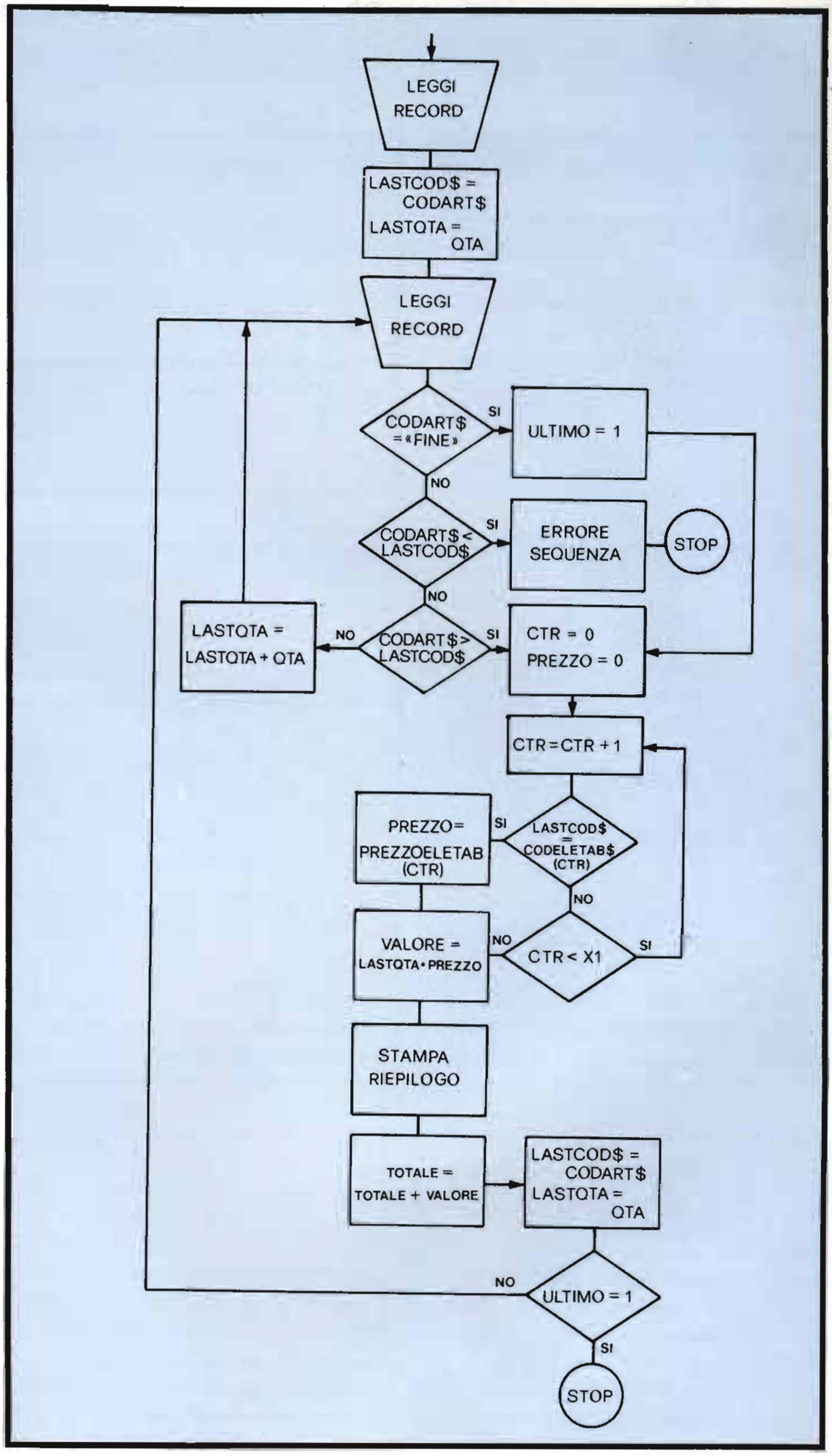


Figura 5. *Diagramma a blocchi riguardante la parte principale del programma. I record del file Movimenti vengono letti ed elaborati uno alla volta sequenzialmente ed i loro codici confrontati con quelli presenti in tabella per ricavare i prezzi unitari.*

Uso delle tabelle con TI99/4A

Come il programma utilizza la tabella

Caricata la tabella in memoria, l'elaborazione del riepilogo inizia alla riga 1120 del programma. Viene letto il primo record e si salvano il numero di codice e la quantità. Si legge quindi il secondo record. Se il codice contiene la parola FINE si elabora l'unico dato letto ed il programma termina. Se il nuovo numero di codice è inferiore al precedente viene inviato il messaggio ERRORE SEQUENZA ed il programma termina. Se il numero di codice è lo stesso, si somma la nuova quantità alla precedente e si legge il record successivo. Infine, se l'ultimo codice letto è maggiore del precedente significa che c'è stata una "rottura di codice" e prima di leggere un altro record bisogna calcolare e stampare il valore relativo al primo articolo. In questo caso il programma passa alla linea 1250 dove inizia la ricerca nella tabella del prezzo unitario. La variabile CTR (contatore) costituisce l'indice dell'elemento della tabella. Si confronta il codice dell'articolo movimentato con il primo codice presente in tabella, poi con il secondo, con il terzo, e così via (linea 1280 del programma). Se i due codici risultano uguali il prezzo unitario sarà l'elemento di tabella PREZZOELETAB (CTR) con lo stesso indice.
Se un codice del file Movimenti non c'è in tabella, il prezzo unitario viene posto uguale a zero e la elaborazione continua. La figura 5 mostra il diagramma a blocchi relativo a questa parte del programma.

TAVOLA DI CONVERSIONE

TI BASIC	BASIC COMMODORE 64
CALL CLEAR	PRINT "(CLR/HOME)"
CALL KEY (0, K, S) IF S = 0 THEN ...	GET A$ IF A$ = " " THEN ...
K (K memorizza direttamente il numero di codice ASCII del tasto premuto)	K = ASC (A$)
IF (K < 49) + (K > 54) THEN ... (la funzione OR è disponibile solo in TI EXT. BASIC)	IF K < 49 OR K > 54 THEN ...
OPEN # 4: "CS1", INPUT, SEQUENTIAL, INTERNAL, FIXED	OPEN 4, 1, 0, "nome del file"
OPEN # 4: "CS1", INPUT, SEQUENTIAL, INTERNAL, FIXED	OPEN 4, 1, 1, "nome del file"
CTR (il TI BASIC considera tutti i numeri come reali)	CT% (il BASIC Commodore distingue le variabili numeriche intere)
TOTALE (il TI BASIC accetta nomi di variabili lunghi fino a 15 caratteri, tutti identificati)	T (il BASIC Commodore identifica solo i primi due caratteri delle variabili. TOTALE non si può usare perché viene identificata come la parola riservata TO)
lo stesso dicasi per ...	
PREZZO	PZ
COD$	CO$
CODELETAB$ (100)	CT$ (100)
PREZZOELETAB (100)	PT (100)
LASTCOD$	LC$
LASTQTA	LQ
Display su 28 colonne.	Display su 40 colonne. Si possono modificare le funzioni TAB.

Figura 6. *Elementi di confronto per la conversione nel BASIC del VIC 20 e del Commodore 64.*

Descrizione del listato

100-240 Menu e selezione opzione.
250-380 Creazione e registrazione della tabella su nastro. la particolare funzione TAB di linea 300 serve ad allineare i numeri a destra.
390-540 Lettura e stampa della tabella per controllarne il contenuto, soprattutto l'ordine crescente dei numeri di codice.
550-700 Creazione file movimenti. La fase termina quando si digita FINE alla richiesta del codice.
710-860 Stampa del file Movimenti per il controllo prima dell'elaborazione principale. L'ultimo record contenente la parola FINE non viene stampato (linea 770).
870 Inizia la parte principale per l'elaborazione del riepilogo.
880 Dimensionamento della tabella. Un aumento indiscriminato dell'indice provoca il messaggio d'errore MEMORY FULL e l'arresto del programma.
890-940 Apertura del file ed indicazioni sull'uso del registratore.
950-1010 Caricamento della tabella in memoria. L'ultimo elemento, contenente la parola FINE, non viene ovviamente caricato.
1020-1110 Inizializzazione di alcune variabili, apertura del file movimenti e stampa della testata.
1120-1240 Lettura dei record, confronto tra un codice ed il successivo, rivelazione dell'ultimo record e di errori di sequenza.
1250-1300 Consultazione della tabella per ricavare il prezzo unitario.
1310-1370 Stampa di una riga di riepilogo.
1380-1460 Stampa del totale, chiu-

Uso delle tabelle con TI99/4A

sura del riepilogo e chiusura del file Movimenti.
1470-1480 Fine del programma.

Variabili usate nel programma

K Contiene il numero di codice ASCII del tasto premuto.
CTR Viene usata come contatore in varie fasi.
COD$ Numero di codice (alfanumerico) dell'articolo.
PREZ Prezzo unitario.
CODART$ Codice articolo letto dal file movimenti.
QTA Quantità.
CODELETAB$ () Elemento di tabella consistente nel codice dell'articolo.
PREZZOELETAB () Elemento di tabella consistente nel prezzo unitario.
TOTALE Valore complessivo degli articoli movimentati.
X Contatore degli elementi di tabella.
X1 Numero degli elementi in tabella.
ULTIMO Variabile flag; il suo valore viene posto uguale ad uno quando è stato letto l'ultimo record del file Movimenti.
LASTCOD$ Ultimo codice letto prima dell'attuale.
LASTQTA Ultima quantità letta prima dell'attuale.
PREZZO Prezzo unitario desunto dalla consultazione della tabella.
VALORE Quantità per prezzo unitario. ■

Listato 1. *Il programma per la creazione e l'uso di tabelle.*

```
100 REM   PROVA USO TABELLA
110 REM   ******************

120 REM  versione: TI BASIC
130 CALL CLEAR
140 PRINT "1. CREA TABELLA"
150 PRINT "2. STAMPA TABELLA
"
160 PRINT "3. CREA FILE MOVI
MENTI"
170 PRINT "4. STAMPA FILE MO
VIMENTI"
180 PRINT "5. STAMPA RIEPILO
GO"
190 PRINT "6. FINE PROGRAMMA
"::::
200 CALL KEY(0,K,S)
210 IF S=0 THEN 200
220 IF (K<49)+(K>54)THEN 200
230 K=K-48
240 ON K GOTO 250,390,550,71
0,870,1470
250 CALL CLEAR
260 OPEN #4:"CS1",OUTPUT,SEQ
UENTIAL,INTERNAL,FIXED
270 CALL CLEAR
280 CTR=0
290 CTR=CTR+1
300 PRINT TAB(4-LEN(STR$(CTR
)));CTR;
310 INPUT "CODICE: ":COD$
320 IF COD$="FINE" THEN 340
330 INPUT "     PREZZO: ":PR
EZ
340 PRINT #4:COD$,PREZ
350 PRINT
360 IF COD$<>"FINE" THEN 290
```

Seguito listato 1.

```
370 CLOSE #4
380 GOTO 100
390 CALL CLEAR
400 OPEN #4:"CS1",INPUT ,SEQ
UENTIAL,INTERNAL,FIXED
410 CALL CLEAR
420 PRINT "TABELLA"
430 PRINT
440 INPUT #4:COD$,PREZ
450 IF COD$="FINE" THEN 480
460 PRINT "COD.: ";COD$,"PRE
ZZO:";TAB(28-LEN(STR$(PREZ))
);PREZ
470 GOTO 440
480 PRINT
490 PRINT " premi space per
continuare"
500 CALL KEY(0,K,S)
510 IF S=0 THEN 500
520 CALL CLEAR
530 CLOSE #4
540 GOTO 100
550 CALL CLEAR
560 OPEN #3:"CS1",OUTPUT,SEQ
UENTIAL,INTERNAL,FIXED
570 CALL CLEAR
580 PRINT "MOVIMENTI"
590 PRINT
600 CTR=0
610 CTR=CTR+1
620 PRINT TAB(4-LEN(STR$(CTR
)));CTR;
630 INPUT "CODICE: ":CODART$
640 IF CODART$="FINE" THEN 670
650 INPUT "  QUANTITA': " :QTA
660 PRINT
670 PRINT #3:CODART$,QTA
680 IF CODART$="FINE" THEN 6
```

Uso delle tabelle con TI99/4A

Seguito listato 1.

```
90 ELSE 610
690 CLOSE #3
700 GOTO 100
710 CALL CLEAR
720 OPEN #3:"CS1",INPUT ,SEQ
UENTIAL,INTERNAL,FIXED
730 CALL CLEAR
740 PRINT "FILE MOVIMENTI"
750 PRINT
760 INPUT #3:CODART$,QTA
770 IF CODART$="FINE" THEN 8
00
780 PRINT "CODICE: ";CODART$
,"QUANTITA':";TAB(29-LEN(STR
$(QTA)));STR$(QTA)
790 GOTO 760
800 PRINT
810 PRINT " premi space per
continuare"
820 CALL KEY(0,K,S)
830 IF S=0 THEN 820
840 CALL CLEAR
850 CLOSE #3
860 GOTO 100
870 CALL CLEAR
880 DIM CODELETAB$(100),PREZ
ZOELETAB(100)
890 PRINT "* CARICA FILE TAB
ELLA"
900 OPEN #3:"CS1",INPUT ,SEQ
UENTIAL,INTERNAL,FIXED
910 TOTALE=0
920 X=0
930 PRINT
940 PRINT "* RUNNING ..."
950 INPUT #3:COD$,PREZ
960 IF COD$="FINE" THEN 1010
970 X=X+1
980 CODELETAB$(X)=COD$
990 PREZZOELETAB(X)=PREZ
1000 GOTO 950
1010 CLOSE #3
1020 X1=X
1030 ULTIMO=0
1040 PRINT
1050 PRINT "* CARICA FILE MO
VIMENTI"
1060 OPEN #5:"CS1",INPUT ,SE
QUENTIAL,INTERNAL,FIXED
1070 CALL CLEAR
1080 PRINT TAB(10);"RIEPILOGO"
```

Seguito listato 1.

```
1090 PRINT ::
1100 PRINT "CODICE  QTA  PRE
ZZO   VALORE"
1110 PRINT "------------------
--------------"
1120 INPUT #5:CODART$,QTA
1130 LASTCOD$=CODART$
1140 LASTQTA=QTA
1150 INPUT #5:CODART$,QTA
1160 IF CODART$<>"FINE" THEN
 1190
1170 ULTIMO=1
1180 GOTO 1250
1190 IF CODART$>=LASTCOD$ TH
EN 1220
1200 PRINT TAB(6);"ERRORE SE
QUENZA"
1210 GOTO 1450
1220 IF CODART$>LASTCOD$ THE
N 1250
1230 LASTQTA=LASTQTA+QTA
1240 GOTO 1150
1250 CTR=0
1260 PREZZO=0
1270 CTR=CTR+1
1280 IF LASTCOD$=CODELETAB$(
CTR)THEN 1300
1290 IF CTR<X1 THEN 1270 ELS
E 1310
1300 PREZZO=PREZZOELETAB(CTR
)
1310 VALORE=LASTQTA*PREZZO
1320 PRINT LASTCOD$;TAB(11-L
EN(STR$(LASTQTA)));LASTQTA;T
AB(18-LEN(STR$(PREZZO)))
;PREZZO;
1330 PRINT TAB(29-LEN(STR$(V
ALORE)));STR$(VALORE)
1340 TOTALE=TOTALE+VALORE
1350 LASTQTA=QTA
1360 LASTCOD$=CODART$
1370 IF ULTIMO<>1 THEN 1150
1380 PRINT "------------------
--------------"
1390 PRINT TAB(14);"TOTALE";
TAB(29-LEN(STR$(TOTALE)));ST
R$(TOTALE)
1400 PRINT "------------------
--------------"
1410 PRINT " premi space per
 continuare"
1420 CALL KEY(0,K,S)
1430 IF S=0 THEN 1420
1440 CALL CLEAR
1450 CLOSE #5
1460 GOTO 100
1470 CALL CLEAR
1480 END
```

Impariamo il linguaggio macchina con il VIC e il C 64

— Parte quarta—

Indirizzamento con puntatore e salti condizionati

di *Alessandro Guida*

Le possibilità di indirizzamento del 6502

La scorsa volta abbiamo visto tre possibili maniere di trattare i dati da parte del 6502, dette modi d'indirizzamento, e come con il tipo d'indirizzamento cambiava il codice dell'istruzione. L'esempio illustrava l'istruzione LDA per caricare un dato nell'accumulatore.
I modi d'indirizzamento già visti sono quelli sottoelencati.
1) Indirizzamento immediato:
LDA # $0A A9 0A
Carica nell'accumulatore il dato riportato dopo l'istruzione LDA.
2) Indirizzamento in pagina zero:
LDA $FE A5 FE
Carica nell'accumulatore il contenuto della locazione in pagina zero specificata dopo LDA.
3) Indirizzamento assoluto:
LDA $10D5 AD D5 1D
Carica nell'accumulatore il contenuto di una qualsiasi locazione di memoria. L'indirizzo di questa locazione, dopo l'istruzione LDA, va depositato in memoria secondo lo standard del 6502 byte-basso/byte-alto.
L'istruzione LDA rende molto intuitivo il concetto di indirizzamento, ma quanto appena detto, e quello che seguirà, è parimenti valido per le altre istruzioni del 6502. Ad esempio, ADC $FE sommerà all'accumulatore il contenuto della locazione $FE in pagina zero. Mentre

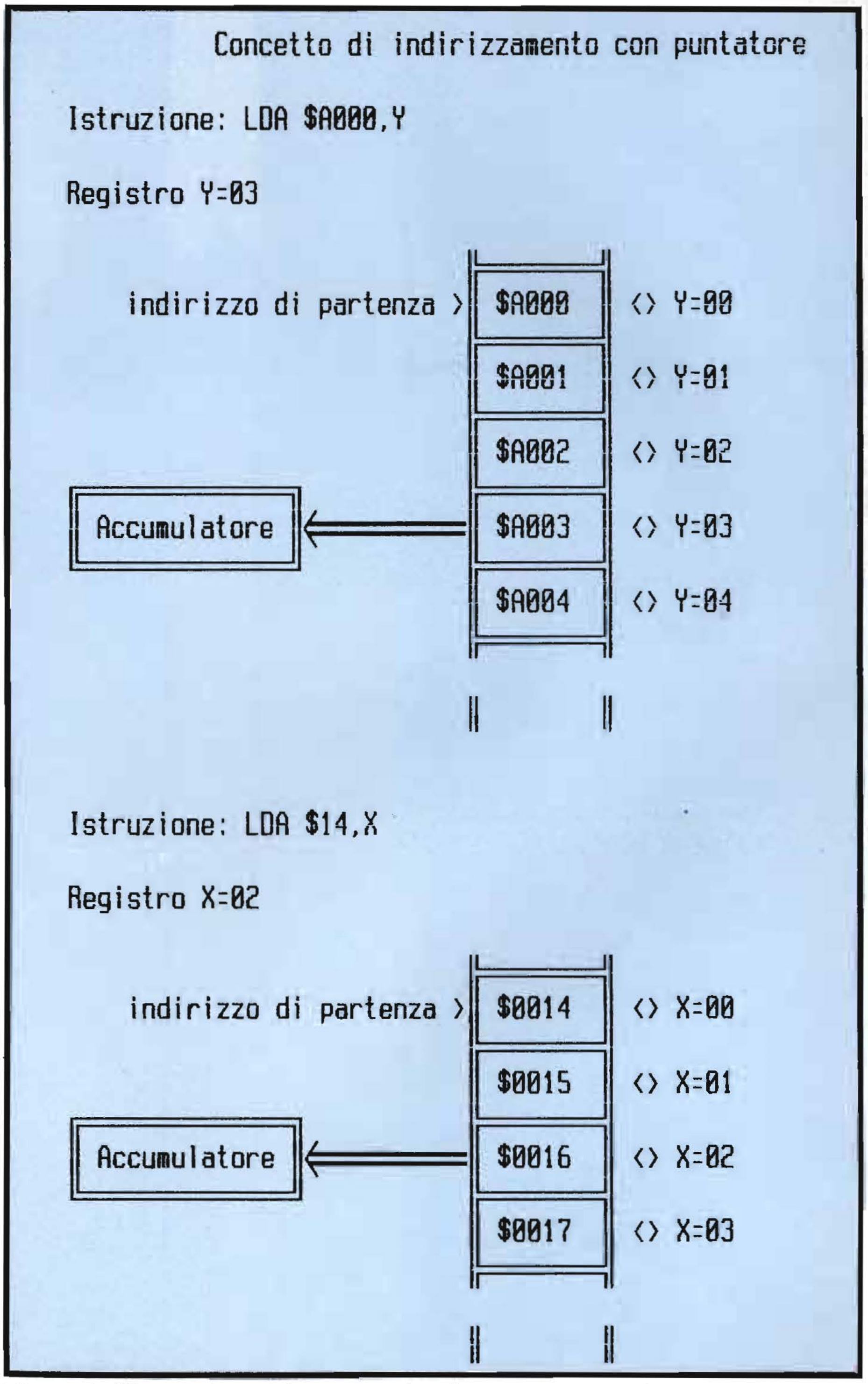


Figura 1. *Illustrazione del concetto di indirizzamento assoluto, Y e in pagina 0, X.*

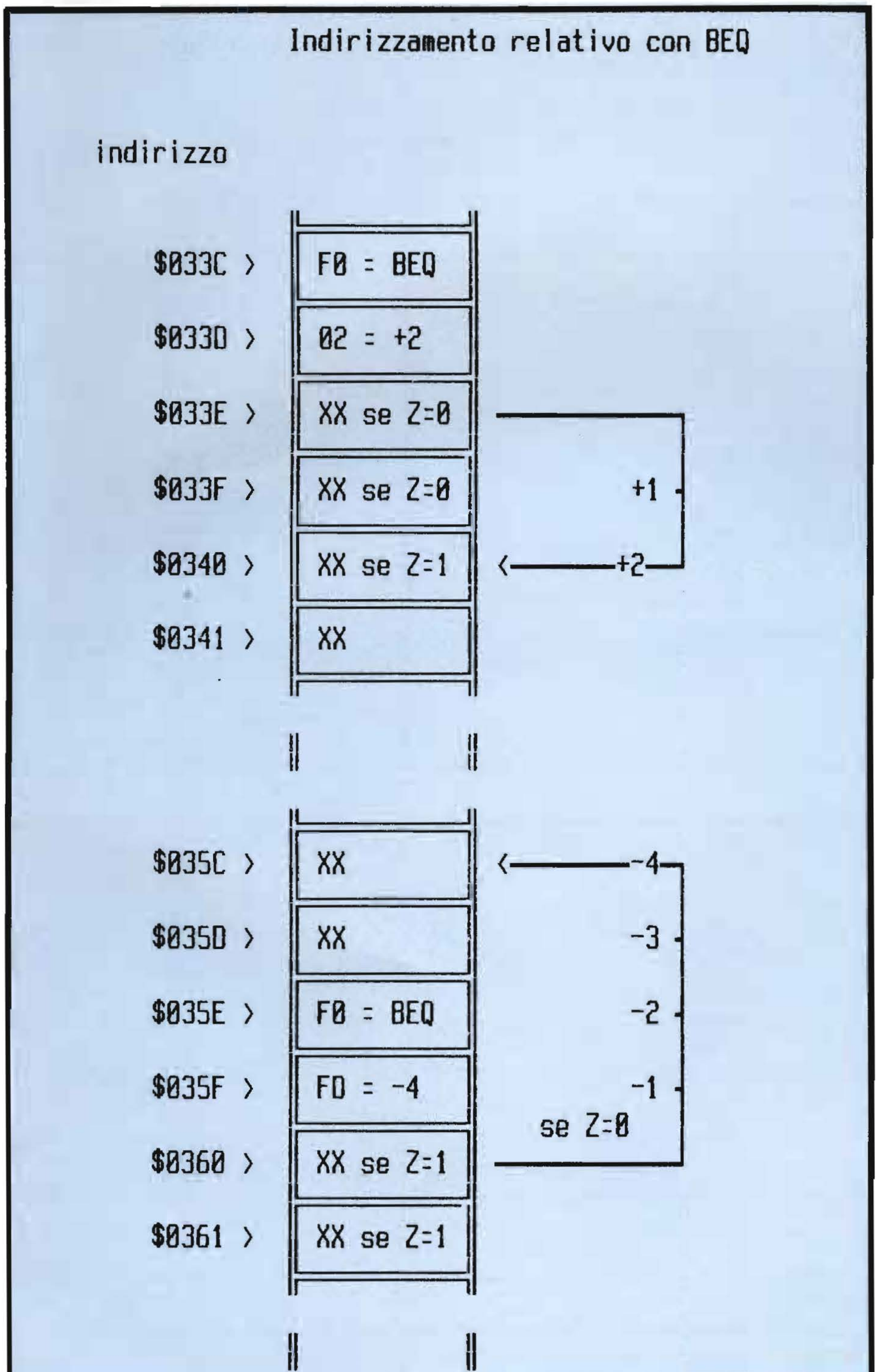


Figura 2. *Illustrazione dell'esecuzione di diramazioni, o salti condizionati, sia positivi che negativi.*

ADC#$0A sommerà all'accumulatore direttamente il numero 10.
Così anche per STA dove, per fare un altro esempio, STA $10D5 memorizza il contenuto dell'accumulatore nella locazione $10D5. Naturalmente, per STA l'indirizzamento immediato non è consentito.
Vedremo di volta in volta gli indirizzamenti consentiti per ogni tipo di istruzione.

Proseguiamo con i tipi di indirizzamento possibili con LDA.

4) Indirizzamento assoluto, X:
LDA $A000, X   BD 00 A0

Carica nell'accumulatore il contenuto della locazione puntata dall'indirizzo assoluto specificato + il contenuto del registro X.

5) Indirizzamento assoluto, Y:
LDA $A000, Y   B9 00 A0

Come per l'assoluto, X. All'indirizzo assoluto viene però sommato il contenuto di Y.

6) Indirizzamento in pagina zero, X:
LDA $45, X   B5 45

Come per l'assoluto, X. Però al posto di un indirizzo assoluto ne è presente uno in pagina zero. Ad esempio, LDA $03, X, se X contiene $04, caricherà in memoria la locazione $07 (3 + 4).
L'uso di questi tre tipi di indirizzamento è evidente: si può tenere fisso un indirizzo base e accedere alle locazioni seguenti, semplicemente cambiando il registro X o Y. Sono quindi l'ideale per la gestione di tabelle e per la realizzazione di cicli.
Naturalmente, la tabella dovrà essere ristretta a 255 elementi, poiché i registri interni del 6502 sono a soli 8 bit.
Se la tabella risiede in pagina zero o 1, si può utilizzare l'indirizzamento in pagina zero, X, altrimenti si deve ricorrere a quello assoluto, X o Y.
In figura 1 sono riportati due esempi di indirizzamento con puntatore.

OPERAZIONE	REG.X	FLAG Z	FLAG N
INX	252	0	1
INX	253	0	1
INX	254	0	1
INX	255	0	1
INX	0	1	0
INX	1	0	0
INX	2	0	0
.	.	.	.
.	..	.	.
.	.	.	.
INX	126	0	0
INX	127	0	0
INX	128	0	1
DEX	5	0	0
DEX	4	0	0
DEX	2	0	0
DEX	1	0	0
DEX	0	1	0
DEX	255	0	1
DEX	254	0	1
INX	255	0	1
INX	0	1	0

Figura 3. *Questa figura mostra l'influenza delle due istruzioni INX e DEX sul registro X e sui flag. L'esempio resta valido anche per il registro Y e le istruzioni INY e DEY.*

```
Tabella codici istruzioni

Istruzione: LDA - Carica accumulatore
            LDX - Carica registro X
            LDY - Carica registro Y

FLAG alterati:        N V B D I Z C
                      X         X

Modo indirizzamento               Codici
                                  LDA LDX LDY

Implicato
Accumulatore
Assoluto                          AD  AE  AC
Pagina-0                          A5  A6  A4
Immediato                         A9  A2  A0
Assoluto,X                        BD      BC
Assoluto,Y                        B9  BE
(IND,X)                           A1
(IND),Y                           B1
Pagina-0,X                        B5      B4
Pagina-0,Y                            B6
Relativo
Indiretto

----------------------------------------------------------

Istruzione: ADC - Somma con Carry

FLAG alterati:        N V B D I Z C
                      X X       X X

Modo indirizzamento               Codice
                                  ADC

Implicato
Accumulatore
Assoluto                          6D
Pagina-0                          65
Immediato                         69
Assoluto,X                        7D
Assoluto,Y                        79
(IND,X)                           61
(IND),Y                           71
Pagina-0,X                        75
Pagina-0,Y
Relativo
Indiretto
```

Tabella 1. *Riepilogo istruzioni di caricamento e somma con riporto.*

Altri comandi del 6502

A questo punto, per poter operare gli indirizzamenti appena visti abbiamo bisogno di nuove istruzioni, che permettono di manovrare i registri X e Y.
L'istruzione per caricare un numero o il contenuto di una cella di memoria, in questi due registri è molto simile al LDA ed è: LDX per il registro X e LDY per il registro Y.
Il comando sarà seguito da un numero, se si vuole caricare un dato in modo immediato, o da una locazione di memoria sia in pagina zero che assoluta o anche sfruttando uno dei tipi di indirizzamento permessi e riportati in tabella 1.
La stessa cosa vale per l'operazione inversa, cioè la memorizzazione dei dati contenuti nei due registri. L'equivalente dello STA per l'accumulatore è STX per il registro X e STY per il registro Y.
Esiste anche una serie di comandi che permette il trasferimento diretto dei dati da un registro all'altro.

Impariamo il linguaggio macchina con il VIC e il C 64

Eccoli:
TAY Trasferisce l'accumulatore nel registro Y.
TAX Trasferisce l'accumulatore nel registro X.
TYA Trasferisce il registro Y nell'accumulatore.
TXA Trasferisce il registro X nell'accumulatore.
NOTA. Nel trasferimento dei dati da un registro all'altro il registro di provenienza non viene alterato. Se, ad esempio, l'accumulatore contiene 6 e operiamo un TAY, dopo il comando il registro Y conterrà 6 e l'accumulatore pure.
Vi sono poi dei comandi che eseguono delle operazioni direttamente sui registri (figura 3):
INY Incrementa di 1 il registro Y.
INX Incrementa di 1 il registro X.
DEY Decrementa di uno il registro Y.
DEX Decrementa di uno il registro X.
NOTA. Poiché, come abbiamo già visto, i registri X e Y sono a 8 bit il loro valore sarà sempre compreso tra 0 e 255. Per cui, se il registro contiene già 255, un'ulteriore operazione di incremento lo riporta a zero. Viceversa, se si trova a zero, una operazione di decremento lo porta a 255.
Questi comandi formati da un solo byte, che operano esclusivamente sui registri interni al 6502, si dicono comandi impliciti o con *indirizzamento implicato*.
Sono le operazioni eseguite più velocemente dal microprocesore.

Un esempio di applicazione delle nuove istruzioni

La gestione di tabelle ricorre nella soluzione di numerosi problemi. In BASIC una tabella può essere immaginata come un vettore, del quale ogni elemento costituisce un dato della tabella. Se, per esempio, si vuole copiare una tabella, formata da 256 elementi, residente in pagina zero nelle locazioni a partire da $C000, in BASIC si ha:

```
10 IN = 0 : FI = 49152 : REM Indirizzi di partenza e arrivo.
20 X = 0 : REM Puntatore agli elementi.
30 POKE FI + X, PEEK (IN + X) : REM Copia elemento num. X.
40 X = X + 1 : REM Incrementa puntatore.
50 IF X < 256 THEN 30 : REM Ripete se non è finita la tab.
```

Questo programmino, benché molto semplice, impiega circa 3 secondi a svolgere il suo dovere. Se anziché 256 elementi ne avessimo dovuto ricopiare 25000, il tempo necessario a svolgere l'operazione in BASIC sarebbe salito a più di 5 minuti.
Non è difficile riscrivere la stessa routine in linguaggio macchina.

```
033C LDX #$00 : 'Puntatore elementi tab.
033E LDA $00, X : 'Legge elemento in pag. zero
0340 STA $C000, X : 'Lo scrive nella nuova tab.
0343 INX : 'Incrementa puntatore
0344 BNE – 08 : 'Se X <> 0 torna a $033E
0346 RTS : 'Ritorna al BASIC
```

Lo svolgimento della routine in linguaggio macchina è molto simile a quello in BASIC, però svolto in meno della metà di un centesimo di secondo.
Dapprima viene messo il registro X, che funziona da puntatore nelle tabelle, uguale a zero. L'elemento 0 è il primo, mentre il 255 è l'ultimo.
Viene quindi letto l'elemento puntato da X nella pagina zero, dove risiede la tabella sorgente, utilizzando l'indirizzamento pagina zero, X.
Il dato letto viene subito memorizzato nella tabella che inizia in $C000, al posto puntato ancora da X. In questo caso si usa l'indirizzamento assoluto, X.
In questa maniera un elemento viene preso dalla tabella sorgente e messo nel posto corrispondente della tabella di arrivo. Il posto nelle due tabelle è determinato da X (lo stesso discorso si potrebbe fare utilizzando il registro Y come puntatore).
Dopo aver copiato l'elemento puntato da X questo registro viene incrementato. L'operazione di incremento, se si supera il valore di 255, riporta a zero il registro in questione. In questo caso viene settato il flag Z del 6502.
L'istruzione seguente serve proprio a controllare se tale flag è settato. Se il flag Z è basso vuol dire che il registro X non ha ancora superato il valore di 255, e l'istruzione riporta il flusso del programma alla locazione $033E.
In pratica il BNE (Branch Not Equal = Salta se diverso da zero) equivale alla linea BASIC:
IF (ultima operazione) <> 0 THEN ....
Traducendo il listato in codice esadecimale avremo:

```
033C LDX #$00        A2 00
033E LDA $00, X      B5 00
0340 STA $C000, X    9D 00 C0
0343 INX             E8
0344 BNE – 08        D0 F8
0346 RTS             60
```

Questa routine la potrete caricare in memoria utilizzando il programma pubblicato sullo scorso numero di **Personal Software**.
Per convincersi della velocità di esecuzione sarà sufficiente digitare: SYS828.

I flag e le istruzioni di salto condizionato

La maggior parte delle istruzioni

viste finora causano dei cambiamenti nel registro del 6502, contenente i flag.
Ad esempio LDA altera i flag N e Z. Il flag N viene attivato se il numero caricato nell'accumulatore è negativo (diciamo quindi che il flag N dipende dallo stato del bit 7 del dato), mentre il flag Z interviene se il numero è uguale a zero. Lo stesso discorso vale per le istruzioni INX e INY.
Vi è poi una serie di istruzioni che permettono di controllare detti flag. Queste sono istruzioni di salto condizionato, che controllano lo stato di uno dei flag e in base al suo contenuto eseguono o meno un salto ad un determinato indirizzo.
Per ora accenneremo solo a due di queste: BEQ e BNE.
BEQ (Branch EQual = salta se uguale a zero) esegue il salto se il flag Z è attivato. BNE, al contrario, esegue il salto se il flag Z è pulito. Sia BNE che BEQ, e tutte le altre istruzioni di questo genere, nel caso non verifichino la condizione richiesta, proseguono con l'istruzione successiva e non eseguono il salto.
L'indirizzamento effettuato da queste istruzioni è detto *"relativo"*, poiché si esplica non come indirizzo assoluto a cui saltare, ma come incremento (o decremento) dell'indirizzo seguente l'istruzione di salto condizionato.
Facciamo un esempio: se l'istruzione BEQ si trova alla locazione 40000 (in decimale), in 40001 vi sarà il valore del salto, da effettuare, supponiamo + 10.
Quando il programma giungerà alla linea 40000, se il flag Z sarà uguale a 0 l'esecuzione proseguirà con la locazione successiva, cioè la 40002. Altrimenti si avrà il salto alla locazione 40002 + 10.
Poiché le istruzioni di salto condizionato devono essere seguite da un solo byte di indirizzamento relativo, il campo di locazioni a cui è possibile saltare va da + 127 a − 128 rispetto alla locazione successiva all'istruzione.
Per i salti negativi vale la regola del complemento a due.
Quindi si prende il numero al positivo, lo si nega e si aggiunge 1.
La figura 2 esemplifica questi concetti e la tabella 2 fornisce un riepilogo delle istruzioni viste.

■

```
Istruzione: STA - Memorizza l'accumulatore
            STX - Memorizza registro X
            STY - Memorizza registro Y

FLAG alterati:    N V B D I Z C

Modo indirizzamento             Codice
                                STA STX STY

Implicato
Accumulatore
Assoluto                        8D  8E  8C
Pagina-0                        85  86  84
Immediato
Assoluto,X                      9D
Assoluto,Y                      99
(IND,X)                         81
(IND),Y                         91
Pagina-0,X                      95      94
Pagina-0,Y                          96
Relativo
Indiretto
-----------------------------------------------------

Istruzione:  TAX - Trasferisce l'accumulatore nel reg.X
             TAY - Trasferisce l'acc. nel reg.Y
             TXA - Trasferisce il registro X nell'acc.
             TYA - Trasferisce il reg.Y nell'acc.

FLAG alterati:    N V B D I Z C
                  X         X

Modo indirizzamento             Codice
                                TAX TAY TXA TYA

Implicato                       AA  A8  8A  89

NOTA: indirizzamento solo implicato.
-----------------------------------------------------

Istruzione: BEQ - esegue salto se Z=1
            BNE - Esegue salto se Z=0

FLAG alterati:    N V B D I Z C

Modo indirizzamento             Codice
                                BEQ BNE

Relativo                        F0  D0

Nota: indirizzamento soltanto relativo
```

Tabella 2. *Riepilogo istruzioni di memorizzazione, trasferimento tra registri e salto condizionato.*

SERVIZIO SOFTWARE

PERSONAL SOFTWARE



P.S. propone ai propri lettori i dischi o le cassette dei programmi pubblicati. I programmi, provati e garantiti, sono di immediato utilizzo.

P.S. n°	Programma	Sistema	Prezzo	Codice	Supporto
3	La carta del cielo Collisione	Apple II	30.000	1	Disco
2	Editor/Assembler in BASIC	CBM 3032	40.000	3	Disco
4	Interi in precisione multipla Grafica 3D	Apple II	40.000	4	Disco
4	Gioco del calcio	CBM 3032	25.000	5	Disco
5	Pretty printer Shape table	Apple II	30.000	6	Disco
7	Data base modulare	Apple II	25.000	7	Disco
12-13	Wei-ch'i	CBM 3032	20.000	8	Cassetta
14	Tool-Kit	C 64	35.000	9	Cassetta

Per richiedere i programmi in contrassegno, pagando direttamente al postino la cifra indicata, inviare il seguente tagliando

Spedire in busta chiusa a Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 Milano

---

Inviatemi i seguenti nastri e/o dischi con i programmi pubblicati su P.S.

Cod. ☐☐☐☐☐☐ **a L.** ....

Cod. ☐☐☐☐☐☐ **a L.** ....

Cod. ☐☐☐☐☐☐ **a L.** ....

Cod. ☐☐☐☐☐☐ **a L.** ....

Spese postali (contributo fisso) L. 2.000

TOTALE L. ....

che pagherò al postino alla consegna del pacco.

GRUPPO EDITORIALE JACKSON

Cognome ....

Nome ....

Indirizzo ....

CAP ....

Città ....

Firma ....

SHARP

I numeri misteriosi

di Mauro Lenzi

In uno degli articoli precedenti abbiamo scoperto, con un metodo molto semplice, tutte le istruzioni di cui lo Sharp è provvisto, osservando, con molta meraviglia, che ben dodici di queste non sono mai state dichiarate nel manuale di istruzioni di questo computer.

Il procedimento adottato per scoprirle è estremamente elementare: consiste infatti semplicemente nell'andare a modificare il byte corrispondente ad una nota istruzione posta in un programma; cambiando in maniera opportuna questo numero, si possono infatti sostituire a quella determinata istruzione tutte quelle di cui il computer dispone.

Questo è possibile perché, al fine di evitare un inutile spreco di memoria, un programma non è direttamente memorizzato dal sistema operativo lettera per lettera, bensì viene prima "compattato", associando ad ogni istruzione un determinato numero.

È interessante notare che questo metodo di codificazione dei programmi è stato adottato dalla maggioranza dei personal computer e talvolta consente di immettere delle istruzioni illecite in un programma. Ogni linea di un programma, al momento della sua immissione nella memoria, viene analizzata e codificata in maniera compatta col suddetto trucco di trasformare una intera istruzione in un numero. Alcuni computer però, come ad esempio il DAI, prima di memorizzarla verificano anche che non vi siano errori di sintassi, ed in caso affermativo non la accettano. Può però capitare di avere bisogno di utilizzare un'istruzione il cui uso è permesso solo in maniera diretta e quindi non può venire accettata come istruzione di un programma. A prima vista il problema può sembrare irrisolubile, ma, invece, è sufficiente molto spesso immettere detta istruzione già codificata all'interno del programma, così il computer non la analizza e la esegue senza problemi.

A questo punto dobbiamo porci una domanda: come riesce il computer a riconoscere una determinata istruzione ed a codificarla?

Il procedimento è molto semplice: in una certa zona della memoria esiste un buffer in cui sono memorizzate tutte le istruzioni, carattere per carattere secondo il codice ASCII, che sono associate a dei particolari "numeri di riconoscimento". Il significato di questi ultimi varia da calcolatore a calcolatore, ma in ogni caso lo scopo finale è sempre quello di far conoscere al sistema operativo tutto quello che deve sapere su ciascuna istruzione, come ad esempio la sua lunghezza, il suo numero di "compattazione" e l'indirizzo della subroutine in linguaggio macchina corrispondente. Così, ogni volta che introduciamo nel computer una stringa di caratteri il sistema operativo andrà a confrontarla con quelle che ha nel suddetto buffer e andrà ad eseguire la giusta subroutine.

```
10 X = 16384
20 IF PEEK X = 98 AND PEEK (X + 1) = 101 AND
PEEK (X + 2) = 94 THEN BEEP 3 : PRINT X
30 X = X + 1 : GOTO 20
```

Listato 1. *Il programma per la ricerca degli indirizzi di memoria in cui sono posti i numeri corrispondenti all'istruzione RUN.*

```
10 INPUT "INDIRIZZO INIZIALE =" ; X
20 I = PEEK X
30 IF I > 80 AND I < 107 THEN PAUSE CHR$ (I -
16) : GOTO 50
40 PAUSE I
50 X = X + 1 : GOTO 20
```

Listato 2. *Il programma utilizzato per trovare l'inizio e la fine del buffer delle istruzioni.*

Bene, adesso dovrebbe essere chiaro quello che bisogna fare per proseguire la nostra indagine all'interno dello Sharp: dobbiamo trovare dove è situato questo buffer.

Il programma 1 è stato progettato per sondare tutta la memoria dello Sharp alla ricerca di questa zona della memoria. Il principio su cui si basa è il seguente: abbiamo appena detto che in questa parte della memoria sono memorizzate tutte le istruzioni, lettera per lettera; allora ci sarà sicuramente, ad esempio, l'istruzione RUN. Quindi possiamo essere certi che da qualche parte della memoria ROM esistono tre indirizzi consecutivi in cui sono presenti i numeri corrispondenti, rispettivamente, alle lettere "R", "U" e "N".

La prima cosa che viene da pensare è che questi numeri siano espressi in codice ASCII, come accade in quasi tutti i computer, ma tuttavia abbiamo già avuto modo di constatare sul nostro Sharp quanto siano originali i giapponesi e quindi credo che nessuno si meraviglierà se le cose non sono esattamente come potevamo aspettarci. Questa volta, però, siamo stati più in gamba di loro, infatti i numeri corrispondenti li conosciamo già! In uno degli articoli precedenti abbiamo visto un programma che ci permetteva di visualizzare tutti i caratteri del nostro computer, scoprendone tra l'altro alcuni molto strani. Con quel programma si vede immediatamente che il numero corrispondente alla "R" è il 98, alla "U" è il 101 ed

alla "N" il 94, che non hanno proprio niente in comune con i numeri del codice ASCII.
Un'ultima possibilità che si deve considerare, prima di eseguire il programma-sonda, è la sfortunata coincidenza di trovare nella ROM tre byte consecutivi uguali ai nostri tre numeri, ma che vi corrispondono solo casualmente. Tuttavia questa probabilità è estremamente bassa e, in ogni caso, ci si accorgerebbe molto facilmente dell'errore osservando che i numeri successivi non corrispondono ad altre istruzioni.

Vediamo finalmente che cosa succede a far girare il nostro primo programma: dopo qualche minuto di attesa si udiranno tre beep ed il display visualizzerà il numero 17173. Abbiamo così trovato quello che stavamo cercando, ma dobbiamo fare ancora una cosa, cioè delimitare con precisione l'area del buffer, perché non sappiamo in che posizione sia posta l'istruzione RUN rispetto a questo. Infatti niente deve farci ritenere che il comando RUN sia posto all'inizio del buffer oppure alla fine.
Per circoscrivere la zona di memoria che ci interessa utilizzeremo il programma 2, che funziona nel modo seguente: dato un certo indirizzo iniziale, il computer cerca di interpretare come lettere i byte corrispondenti a quell'indirizzo, ed ai seguenti, purché siano numeri compresi fra 81 e 106, che sappiamo corrispondere alle lettere dell'alfabeto.
Dando come indirizzo iniziale il nostro 17173, comparirà sul display una lunga sequenza di lettere e numeri che inizierà così:
R U N 176 I 122 198 R E T U R N 222 H 30 164 R E A D. Vedremo così sfilare sotto i nostri occhi una buona parte delle istruzioni dello Sharp, fino a quando non inizieranno a comparire lettere e numeri senza senso, segno indiscutibile della fine del buffer. Per trovarne l'inizio dovremo dare invece degli indirizzi di partenza inferiori a 17173. Con qualche tentativo si potrà verificare che l'indirizzo iniziale è 16683, mentre quello finale è 17377.

La prossima volta commenteremo lungamente queste nuove scoperte e vedremo che si riveleranno estremamente utili; per il momento mi limito a fare qualche piccola osservazione. Innanzitutto, come era da aspettarsi, troviamo normalmente elencate in questa lunga lista anche tutte quelle istruzioni stranissime che avevamo scoperto in precedenza.
Cosa abbastanza curiosa è che siano disposte in una specie di ordine alfabetico, forse per permettere al sistema operativo un confronto più rapido. Infine, per chiudere con un po' di suspence, fra una istruzione e la successiva si trovano quattro numeri "misteriosi", il cui segreto verrà svelato, almeno in parte, la prossima volta e ci permetteranno di giungere fino al "cuore" di questo computer.

TEXAS TI99/4A

Gestione file con un solo registratore

di Sergio Borsani

Uno dei compiti principali ai quali può essere adibito un elaboratore è quello di memorizzare e tenere aggiornati dei dati, utilizzando le così dette "memorie di massa", cioè unità di registrazione su nastro o su disco. Le prime hanno l'unico vantaggio di essere molto più economiche; le seconde presentano la caratteristica di poter accedere direttamente ad un dato e sono più veloci.
Spesso mi sono sentito rivolgere la domanda su cosa sia possibile fare nei riguardi della gestione dei file, nella configurazione base, con un solo registratore. Francamente ritengo che le possibilità siano limitate. Con due unità di registrazione le cose andrebbero appena un po' meglio ed è per questa ragione che viene fornito un cavetto con doppi attacchi, uno per lettura/scrittura ed un altro per sola lettura (ho saputo recentemente che gli ultimi cavetti vengono forniti con un'unica uscita).

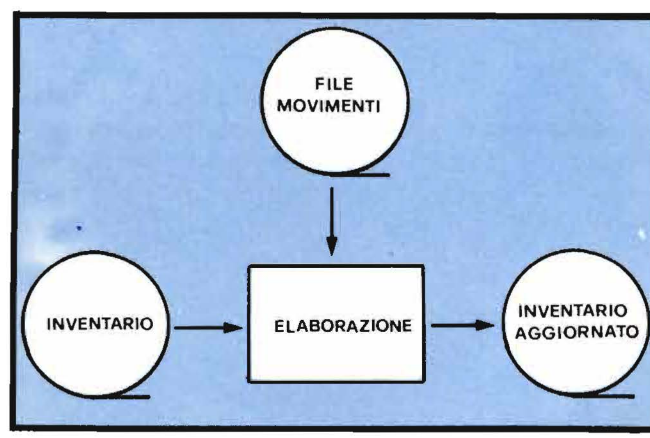


Figura 1. *Flow-chart relativo ad una elaborazione di file sequenziali. I simboli circolari indicano che essi risiedono su nastro magnetico.*

Lo schema principale, e semplificato, in questo genere di elaborazione è illustrato dal flow-chart di figura 1.
È evidente che l'unica via da seguire per chi dispone di un unico registratore è quella di memorizzare interamente il file da aggiornare nella memoria interna del computer (RAM), aggiornare tale file utilizzandone un secondo residente su nastro e quindi, terminate le operazioni di aggiornamento, salvare il file su una porzione libera del nastro, che sia ben identificabile con il numero del tape counter.

I SEGRETI DEI PERSONAL

Gestione file con un solo registratore

```
100 A$="PROVA RECORD"
110 OPEN #3:"CS1",OUTPUT,SEQUENTIAL,FIXED
120 FOR J=1 TO 10
130 PRINT #3:A$
140 NEXT J
150 CLOSE #3
160 DIM R$(10)
170 OPEN #3:"CS1",INPUT ,SEQUENTIAL,FIXED
180 N=0
190 FOR J=1 TO 10
200 INPUT #3:A$
210 N=N+1
220 R$(N)=A$
230 NEXT J
240 CLOSE #3
250 CALL CLEAR
260 FOR J=1 TO 10
270 PRINT R$(J)
280 NEXT J
290 END
```

Listato 1. *Semplice programma utilizzato per scrivere e leggere un file su nastro.*

```
100 A$="PROVA RECORD"
110 OPEN #3:"CS1",OUTPUT,SEQUENTIAL,FIXED
120 FOR J=1 TO 10
130 PRINT #3:A$;J
140 NEXT J
150 CLOSE #3
160 DIM R$(10,2)
170 OPEN #3:"CS1",INPUT ,SEQUENTIAL,FIXED
180 N=0
190 FOR J=1 TO 10
200 INPUT #3:A$,NUMREC
210 N=N+1
220 R$(N,1)=A$
225 R$(N,2)=STR$(NUMREC)
230 NEXT J
240 CLOSE #3
250 CALL CLEAR
260 FOR J=1 TO 10
270 PRINT R$(J,1);" ";R$(J,2)
280 NEXT J
290 END
```

Listato 2. *Lo stesso programma del listato 1, ma con il record composto di 2 campi.*

```
110 OPEN #3:"CS1",OUTPUT,SEQUENTIAL,INTERNAL,FIXED

170 OPEN #3:"CS1",INPUT ,SEQUENTIAL,INTERNAL,FIXED
```

Listato 3. *Linee da modificare per rendere funzionante il programma 2.*

```
100 REM   FILE MOVIMENTI
110 REM ******************
120 CALL CLEAR
130 N=0
140 OPEN #3:"CS1",OUTPUT,SEQUENTIAL,
    INTERNAL,FIXED
150 CALL CLEAR
160 DISPLAY AT(6,1):"RECORD N."
170 DISPLAY AT(11,1):"---- - ---------"
180 DISPLAY AT(12,1):"COD. T QUANTITA'"
190 N=N+1
200 DISPLAY AT(6,10):N
210 CALL HCHAR(10,3,32,28)
220 ACCEPT AT(10,1)SIZE(4):COD$
230 IF COD$="FINE" THEN 260
240 ACCEPT AT(10,6)VALIDATE("EU")SIZE(1):TM$
250 ACCEPT AT(10,8)VALIDATE(DIGIT)SIZE(8):QTA
260 PRINT #3:COD$;TM$;QTA
270 IF COD$<>"FINE" THEN 190
280 CLOSE #3 :: CALL CLEAR :: END
```

Listato 4. *Programma per creare il file Movimenti.*

Il manuale è effettivamente povero di esempi in questo campo, e prima di passare a programmi più complessi è preferibile eseguire alcune esperienze elementari.

L'elemento principale di un file dati è il record. Un file è formato dalla successione di tanti record. Nel listato 1 i record sono tutti uguali e sono rappresentati dalla stringa A$.

Il numero che accompagna l'istruzione OPEN di linea 110 è a discrezione del programmatore, ma deve corrispondere alle successive PRINT #. "CS1" si riferisce all'unità di registrazione; OUTPUT indica che i dati vengono trasferiti dal computer al nastro; inoltre, l'organizzazione può essere solo sequenziale (SEQUENTIAL) ed i record devono essere tutti della stessa lunghezza (FIXED).

Nella seconda parte del programma (linee 160-290) il file viene letto e caricato nella matrice unidimensionale, o variabile con indice, R$ e stampato sul video. Il record, normalmente, non contiene una sola informazione, ma è una struttura che viene suddivisa in più campi, contenenti informazioni elementari tra di loro omogenee.

I campi di un record potrebbero contenere, ad esempio, nome, indirizzo e numero di telefono, oppure il numero di codice, la descrizione di un articolo ed il prezzo unitario, ecc.

Modifichiamo il listato n. 1 alle linee 130, 160, 200, 270 ed aggiungiamo la riga 225. Ora il record è formato da due campi: A$ e J nella linea 130 (listato 2).

Se lanciate il programma, tuttavia, vi accorgerete che c'è qualcosa che non va.

Si potrebbe rimediare aggiungendo una routine che trasformi il campo numerico in variabile stringa, porti i due campi ad una determinata lunghezza, aggiungendo space (CHR$ (32)), ed infine sommi i due campi in modo da ottenere un'unica stringa alfanumerica.

In fase di lettura si dovrebbe decodificare la stringa sapendo, ad esempio, che i primi 20 caratteri formano il primo campo e che i restanti 8 formano il

Gestione file con un solo registratore

```
100 REM   FILE GIACENZE MAGAZZINO
110 REM ***************************
120 CALL CLEAR
130 PRINT TAB(6);"GIACENZE MAGAZZINO":"":"":"":""
140 PRINT "PREMI:":"1. CREA FILE":"2. DISPLAY FILE"
150 PRINT "3. RICHIAMA FILE":"4. AGGIORNA FILE"
160 PRINT "5. REGISTRA FILE":"6. FINE PROGRAMMA":""
170 CALL KEY(0,K,S):: IF S=0 THEN 170
180 IF K<49 OR K>54 THEN 170
190 K=K-48
200 ON K GOTO 210,500,400,590,710,800
210 CALL CLEAR
220 N=0
230 OPEN #3:"CS1",OUTPUT,SEQUENTIAL,INTERNAL,FIXED
240 CALL CLEAR
250 DISPLAY AT(6,1):"RECORD N."
260 DISPLAY AT(11,1):"---- ------------   --------"
270 DISPLAY AT(12,1):"COD. ARTICOLO       GIACENZA"
280 N=N+1
290 DISPLAY AT(6,10):N
300 CALL HCHAR(10,3,32,28)
310 ACCEPT AT(10,1)SIZE(4):COD$
320 IF COD$="FINE" THEN 350
330 ACCEPT AT(10,6)SIZE(12):ART$
340 ACCEPT AT(10,21)VALIDATE(DIGIT)SIZE(8):QTA
350 PRINT #3:COD$
360 IF COD$="FINE" THEN 390
370 PRINT #3:ART$,STR$(QTA)
380 GOTO 280
390 CALL CLEAR :: CLOSE #3 :: GOTO 120
400 CALL CLEAR
410 DIM CAMP$(100,3)
420 OPEN #3:"CS1",INPUT ,SEQUENTIAL,INTERNAL,FIXED
430 CTR=0
440 CTR=CTR+1
450 INPUT #3:CAMP$(CTR,1)
460 IF CAMP$(CTR,1)="FINE" THEN 490
470 INPUT #3:CAMP$(CTR,2),CAMP$(CTR,3)
480 GOTO 440
490 CLOSE #3 :: CALL CLEAR :: GOTO 120
500 CALL CLEAR :: FOR J=1 TO CTR-1
510 PRINT "* RECORD";J;"*"
520 PRINT CAMP$(J,1);TAB(6);CAMP$(J,2);TAB(20);
530 PRINT USING "########":VAL(CAMP$(J,3))
540 PRINT :: NEXT J
550 PRINT
560 PRINT TAB(10);"FINE FILE"
570 PRINT TAB(8);"PREMI UN TASTO"
580 CALL KEY(0,K,S):: IF S=0 THEN 580 ELSE 120
590 CALL CLEAR
600 OPEN #4:"CS1",INPUT ,SEQUENTIAL,INTERNAL,FIXED
610 INPUT #4:COD$,TM$,QTA
620 IF COD$="FINE" THEN 700
630 FOR J=1 TO CTR-1
640 IF COD$<>CAMP$(J,1)THEN 680
650 IF TM$="E" THEN W=1 ELSE W=-1
660 GIACENZA=VAL(CAMP$(J,3))
670 CAMP$(J,3)=STR$(GIACENZA+W*QTA):: GOTO 610
680 NEXT J
690 PRINT "ERRORE CODICE MOVIMENTO";J :: GOTO 610
700 CALL CLEAR :: CLOSE #4 :: GOTO 120
710 CALL CLEAR
720 OPEN #5:"CS1",OUTPUT,SEQUENTIAL,INTERNAL,FIXED
730 N=0
740 N=N+1
750 PRINT #5:CAMP$(N,1)
760 IF CAMP$(N,1)="FINE" THEN 790
770 PRINT #5:CAMP$(N,2);CAMP$(N,3)
780 GOTO 740
790 CALL CLEAR :: CLOSE #5 :: GOTO 120
800 CALL CLEAR :: END
```

Listato 5. *Programma principale che elabora l'inventario aggiornato.*

secondo campo.
Alcuni linguaggi di tipo commerciale, come il COBOL, sono particolarmente adatti alla definizione dei campi e dei record. Ma non è detto che con il BASIC le cose debbano andare poi tanto male. Provate a sostituire le OPEN del listato 2 con quelle del listato 3.
Il manuale dà solo brevi accenni sull'uso dell'opzione INTERNAL, contrapposta all'altra che ha valore di default, DISPLAY. Se fate girare il programma vi convincerete che il formato INTERNAL è quello che fa per noi, in questa circostanza.
Ora che sappiamo registrare e caricare un file, con record formati da più campi, siamo in grado di affrontare un programma più complesso. Immaginiamo di voler aggiornare un file con le giacenze di magazzino (file Inventario), utilizzando i dati di un secondo file (file Movimenti).

Il record dell'inventario è formato da tre campi: codice, descrizione articolo e quantità in giacenza; il record movimento è anch'esso formato da tre campi: codice, tipo movimento (Entrata/Uscita) e quantità. I numeri di codice dei due tipi di record dovranno corrispondere, anche se non tutti gli articoli dell'inventario saranno interessati da variazioni.

Le operazioni dovranno susseguirsi con il seguente ordine: creare i due file con i due distinti programmi riportati nei listati 4 e 5, caricare in memoria il file con le giacenze di magazzino, utilizzando il programma principale (listato 5) e infine, sempre con lo stesso programma, elaborare l'inventario aggiornato.
La matrice che contiene il file Inventario è stata dimensionata a 100. Tale valore può essere aumentato ed è limitato solo dalle capacità di memoria del computer; è tuttavia evidente che in tal modo non potrà mai essere elaborato un file con un numero elevato di record.
Per finire, alcuni brevi commenti.
Si sarebbe potuto utilizzare il TI BASIC, ma si è preferito il TI Extended BASIC per la presenza delle istruzioni DISPLAY AT ed ACCEPT AT (con le opzioni VALIDATE e SIZE), che permettono di creare una "maschera" per l'introduzione dei dati.

Con il registratore non si può usare l'istruzione EOF (End Of File), pertanto bisogna fare in modo che l'ultimo record sia riconoscibile tramite un campo particolare, di solito il primo. Se esso è alfanumerico, si può occupare con la parola "FINE" (soluzione qui adottata), se invece è numerico si riempie, secondo le circostanze, di zeri o di 9.
Attenzione a non sovrapporre programmi e file dati durante le registrazioni!

ZX SPECTRUM

Semplificazione della procedura di IN da tastiera

di Marcello Spero

Questa volta vediamo una routine che, nata per restituire la compatibilità fra il software scritto per le versioni ONE e TWO dello Spectrum e la più recente ISSUE THREE, si è arricchita strada facendo di elementi che possono renderla interessante, anche indipendentemente da questo problema; il suo uso consente infatti di semplificare notevolmente i calcoli richiesti per trovare i valori appropriati da dare come argomento ad una istruzione IN, affinché questa legga lo stato del o dei tasti desiderati. Ma vediamo le cose dall'inizio.

A partire dal Natale scorso è iniziata anche sul nostro mercato la distribuzione della nuova versione dello Spectrum, la ISSUE THREE.

Il motivo che ha spinto i tecnici della Sinclair a sostituire la vecchia ISSUE TWO pare sia duplice: da un lato la necessità di poter montare memorie di varie marche diverse, a seconda della disponibilità (l'ISSUE TWO era legata all'utilizzo di integrati Texas o Texas compatibili), dall'altro i problemi sperimentati con vari tipi di televisore nella corretta riproduzione dei colori e dei caratteri.

La realizzazione di questi obiettivi ha portato ad una macchina senz'altro più affidabile, ma ha anche introdotto qualche leggera differenza di funzionamento, rispetto alle precedenti versioni.

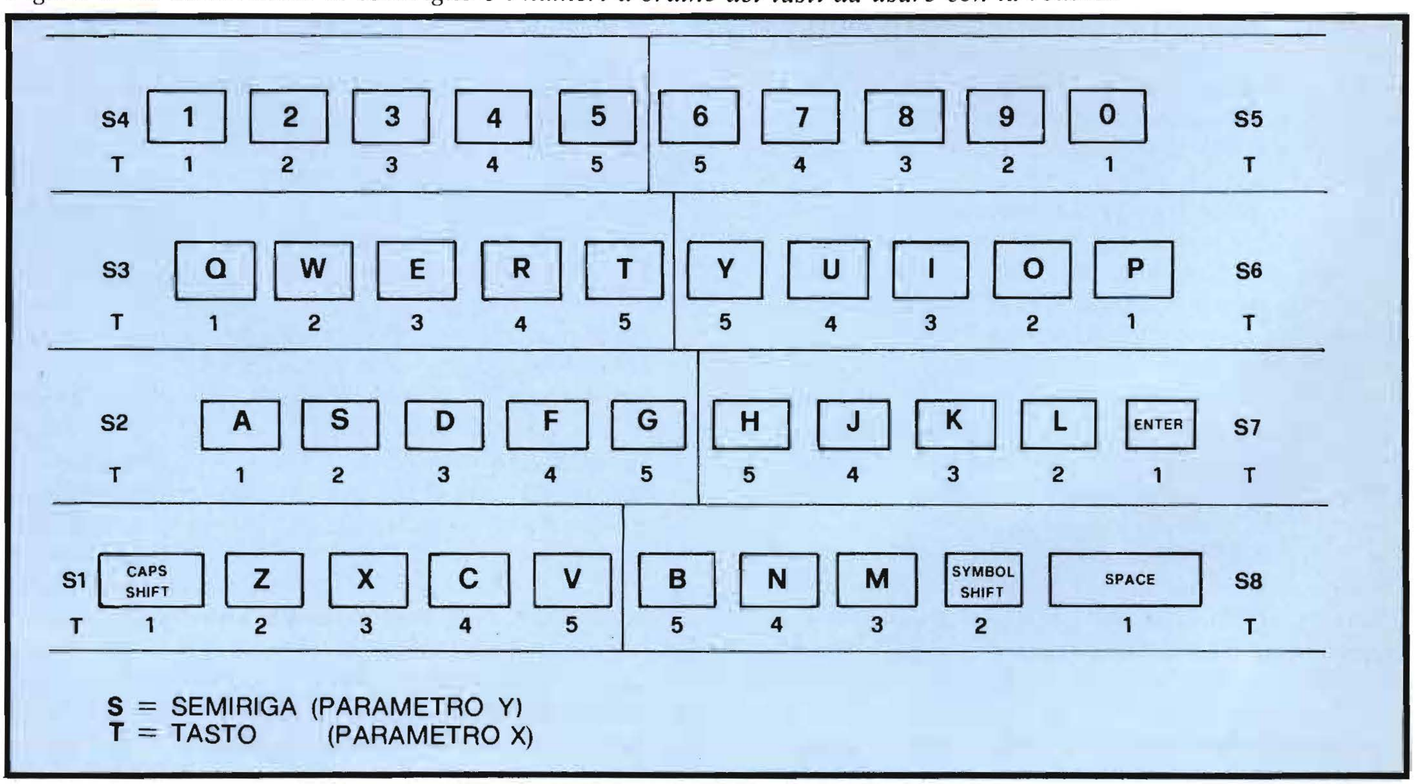
Una di queste differenze, in particolare, coinvolge direttamente il BASIC e provoca una non perfetta compatibilità del software prodotto per le versioni precedenti con la ISSUE THREE.

Per la verità, il problema salta fuori veramente di rado, poiché raramente viene usata l'istruzione che lo provoca: IN.

Senza scendere nei dettagli (chi volesse approfondire può fare riferimento al manuale) ricorderemo che la funzione IN serve a leggere il valore presente in un dato istante su di un certo "port" (canale di ingresso/uscita), specificato come argomento della stessa funzione. Nel caso più normale, quello cioè di uno Spectrum senza hardware aggiuntivo da cui ricevere dati attraverso i port, l'unico uso possibile di IN è la lettura dei segnali provenienti dalla tastiera e dalla presa EAR.

La lettura della tastiera effettuata in questo modo, piuttosto che per mezzo di INKEY$, risulta più immediata e flessibile, permettendo tra l'altro di recepire la pressione anche del solo tasto SHIFT o SIMBOL SHIFT, nonché di due tasti contemporaneamente (vedi **Personal Software** n. 7).

Figura 1. La suddivisione in semirighe e i numeri d'ordine dei tasti da usare con la routine.



Semplificazione della procedura di IN da tastiera



Figura 2. *Struttura del byte dati, letto da ciascun port di tastiera.*

Queste caratteristiche fanno sì che IN venga talvolta usata in programmi di gioco, appunto al posto di INKEY$. E qui iniziano i problemi.

Come vedete dalla figura 2, i dati relativi alla tastiera occupano, per ciascuno dei port interessati (li trovate in figura 3), solo i primi cinque bit. Il settimo bit riporta invece, per tutti i port di tastiera, il valore presente alla presa EAR.

Ciascun tasto è rappresentato, all'interno del proprio port, dal bit corrispondente alla sua posizione rispetto all'esterno della macchina. Il tasto "1" sarà quindi indicato dal bit 0, mentre il "5" dal 4; dal 4 del proprio port sarà indicato la "Y", e dallo 0 il "P".

La prassi corretta per la lettura di un port strutturato in questo modo è l'isolamento del bit che ci interessa e quindi la lettura del suo valore, che sarà 1 se il tasto corrispondente non è premuto, o 0 in caso sia invece premuto. In BASIC, però, una tale operazione è complicata; molti (anche produttori di software commerciale ...) hanno perciò preferito leggere tutti insieme i bit del port come un numero, deducendo poi dal suo valore quali sono i bit a zero e quali ad uno.

Un procedimento di questo tipo può funzionare però solamente se si conoscono i valori assunti da tutti i bit del port nelle diverse situazioni. La pratica aveva insegnato che, sia i bit non utilizzati che quello dedicato alla presa EAR mantenevano un valore costante di UNO, in condizioni normali.

Quindi: nessun tasto premuto, uguale bit 0-4 ad uno; bit 5-7 sempre ad uno, risultato 255 (BIN 11111111) come valore indicante "nessun tasto premuto".

Pare però che le cose non vadano più in questo modo con l'ISSUE THREE; il valore dei port di tastiera, indicante "nessun tasto premuto" sembra sia 191 (BIN 10111111), che indicherebbe come zero il valore normale del bit relativo alla presa EAR.

Il punto non è però l'aver trovato o meno questo nuovo valore, ma avere un sistema che consenta al software scritto in BASIC di essere indipendente dalla versione (ONE, TWO, THREE e chissà, forse FOUR ...) di Spectrum utilizzata, e quindi una lettura con la funzione IN che non tenga conto dei bit che non interessano.

Questo è appunto quanto svolge la routine in linguaggio macchina che vedete nel listato 1.

PORT	TASTI LETTI
65278	CAPS SHIFT - V
65022	A - G
64510	Q - T
63486	1 - 5
61438	0 - 6
57342	P - Y
49150	ENTER - H
32766	SPACE - B

Figura 3. *Elenco dei port e relativi tasti letti.*

Vale la pena innanzitutto di considerare il particolare sistema usato per passare i parametri alla subroutine. Mentre, infatti, il passaggio di dati dal linguaggio macchina al BASIC può avvenire semplicemente attraverso il registro BC dello Z80, il cui contenuto diventa il valore della USR al termine della sua esecuzione, per fornire dati ad una subroutine in linguaggio macchina dal BASIC normalmente è necessario ricorrere a delle POKE.

Questo sistema nel nostro caso è però inaccettabile, poiché fa perdere al software proprio la linearità e semplicità che cerchiamo di restituirgli.

Il sistema adottato si serve invece di una DEF FN, che contiene il salto al linguaggio macchina; verrà richiamata dal BASIC con FN ogni qual volta sarà necessario. Proprio l'uso di queste istruzioni rende possibile un efficiente passaggio di dati alla subroutine: definendo infatti una DEF FN i (a, b) = USR "indirizzo di inizio" e richiamandola con ... FN i (x, y), dove x e y sono particolari valori (nel nostro caso riguardanti il tasto da leggere), questi verranno praticamente "introdotti" nella DEF FN. Al momento immediatamente seguente ogni chiamata l'istruzione DEF FN verrà ad avere la forma:

DEF FN i (x 0E XX XX XX XX XX, y 0E YY YY YY YY YY) = ...

in cui i cinque byte indicati con XX contengono il valore del parametro passato con la x, nella forma (un byte di esponente e quattro di mantissa) spiegata dal manuale nel capitolo riguardante le modalità di memorizzazione dei numeri. La stessa cosa è valida per i cinque byte indicati da YY, i quali contengono il valore del parametro y. In particolare nel nostro caso, essendo i dati, come vedremo, numeri interi e minori di 255, l'unico byte contenente una cifra significativa sarà il terzo di ciascun parametro. Rimane da definire il modo in cui la subroutine potrà accedere a

I SEGRETI DEI PERSONAL

Semplificazione della procedura di IN da tastiera

```
org 50000
equ 23563 DEFADD
;
;
;RECUPERO POS. TASTO
;
50000 2A 0B 5C      ld hl,(DEFADD)
50003 11 04 00      ld de,4
50006 19            add hl,de
50007 46            ld b,(hl)
;
;CONTROLLO VALIDITA'
; VALORE OTTENUTO
;
50008 78            ld a,b
50009 FE 06         cp 6
50011 30 22         jr nc,ERR
;
;
;RECUPERO VALORE SEMIRIGA
;
50013 19            add hl,de
50014 19            add hl,de
50015 4E            ld c,(hl)
;
;
;CONTROLLO VALIDITA'
; VALORE OTTENUTO
;
50016 79            ld a,c
50017 FE 09         cp 9
50019 30 1A         jr nc,ERR
;
;
;CREAZIONE DEL VALORE
;DEL PORT CORRISPONDENTE
;
50021 C5            push bc
50022 1E FE         ld e,254
50024 41            ld b,c
50025 3E FF         ld a,255
50027 B7            or a
LOOP
```

Listato 1. *Assembly della routine.*

```
50028 17            rla
50029 10 FD         djnz LOOP
50031 57            ld d,a
;
;
;LETTURA DELLA TASTIERA
;
50032 D5            push de
50033 C1            pop bc
50034 ED 78         in a,(c)
;
;
;ISOLAMENTO BIT VOLUTO
;
50036 C1            pop bc
LOOP2
50037 1F            rra
50038 10 FD         djnz LOOP2
;
;
;CREAZIONE DEL VALORE
; DA PASSARE ALLA FN
;
50040 3F            ccf
50041 3E 00         ld a,0
50043 CE 00         adc a, 0
50045 4F            ld c,a
50046 C9            ret
;
;
;ROUTINE DI ERRORE
;PRODUZIONE DEL SUONO
;E DEL VALORE PER LA FN
;
ERR
50047 1E FF         ld e,255
50049 21 C8 00      ld hl,&00C8
50052 CD B5 03      call &03B5
50055 01 02 00      ld bc,2
50058 C9            ret
;
;
;
```

Seguito listato 1.

questi dati. La maniera più semplice sfrutta il metodo usato normalmente dal sistema per recuperare i parametri delle normali FN. Non appena viene incontrata una istruzione che contiene una FN valida (cioè in armonia per nome e per numero e tipo dei parametri con una DEF FN esistente all'interno del programma) la variabile di sistema DEFADD (23563) viene fatta puntare al nome della prima delle variabili costituenti i parametri all'interno della DEF FN interessata. Basterà a questo punto incrementare di 4 tale valore per ottenere l'indirizzo del terzo byte della variabile stessa, che è appunto quello che ci interessa. Successivamente incrementeremo questo valore di 8 per avere l'indirizzo del terzo byte della seconda variabile.

Tutto questo si può esprimere in BASIC come:

```
LET a = PEEK 23563
LET a = a + 4
...
LET a = a + 8
```

e in Assembly come:

```
LD HL, (23563)
LD DE, 4
ADD HL, DE
...
ADD HL, DE
ADD HL, DE
```

Questo metodo può essere utilmente applicato a qualsiasi routine in linguaggio macchina, che necessiti di dati provenienti dal BASIC.

Vediamo adesso il funzionamento della routine vera e propria.

L'istruzione (per l'esattezza si tratta di una funzione) IN viene usata nella forma:

IN p

dove p è il numero che contraddistingue il port da cui vogliamo leggere. Per conoscere lo stato di un particolare tasto dovremmo a questo punto eseguire dei controlli sul valore ottenuto, controlli che, come abbiamo visto, sono possibili solo a patto di conoscere il valore assunto dal port quando non è premuto alcun tasto.

Usando invece la nostra routine potremo sostituire a questo procedimento una istruzione:

LET a = FN i (x, y)

dove x sarà il numero d'ordine del tasto partendo dall'esterno, per ciascuna semiriga di cinque, e y sarà il numero d'ordine della semiriga. Tutto questo è più

```
 42, 11, 92, 17,  4,  0, 25, 70,
120,254,  6, 48, 34, 25, 25, 78,
121,254,  9, 48, 26,197, 30,254,
 65, 62,255,183, 23, 16,253, 87,
213,193,237,120,193, 31, 16,253,
 63, 62,  0,206,  0, 79,201, 30,
255, 33,200,  0,205,181,  3,  1,
  2,  0,201
```

Listato 2. *Codice macchina della routine.*

```
   1 REM
********************************
*                              *
*      PROGRAMMA DI PROVA      *
*   PER LA ROUTINE DI LETTURA  *
*         DELLA TASTIERA       *
*                              *
********************************
  10 DEF FN i (a,b) =USR 50000
  20 INPUT "n. tasto",a
  30 INPUT "n. semiriga",b
  40 PAUSE 0
  50 PRINT FN i (a,b)
  60 GO TO 20
```

Listato 3. *Programma di prova.*

chiaro nella figura 1.

Di ritorno otterremo nella variabile a il valore 1 se il tasto è premuto, o zero in caso contrario. Se, infine, avremo commesso qualche errore nel passare i parametri (x deve essere compreso fra 1 e 5, y fra 1 e 8) otterremo di ritorno un valore 2, accompagnato da un breve segnale sonoro.

Oltre a semplificare i conti, questo sistema ha il vantaggio di isolare ciascun tasto dagli altri; in altre parole, otterremo una risposta corretta riguardo ad un tasto, indipendentemente dal fatto che contemporaneamente siano premuti altri tasti che fanno capo allo stesso port.

Veniamo ora ad un esempio pratico. Supponiamo di voler sapere se è premuto il tasto corrispondente al cursore verso destra, cioè l'"8"; scriveremo:

DEF FN i (a, b) = USR v

dove v è l'indirizzo di inizio della routine, in un punto qualsiasi del programma, e quindi:

LET a = FN i (3, 5)
IF a = 1 THEN ...

o, più semplicemente:

IF FN i (3, 5) THEN ...

Nel caso invece i parametri vengano determinati dal programma, e sia quindi possibile un errore, è opportuno inserire un controllo per il caso di FN i (x, y) = 2. Letture combinate di più tasti andranno fatte mediante più chiamate della FN i con valori diversi, eventualmente correlate in una:

IF ... AND ... THEN ...

Nel listato 1 trovate l'Assembly della routine, mentre nel listato 2 c'è il codice macchina già assemblato, pronto da introdurre con uno dei numerosi programmini (questo nome è anche troppo per loro) di caricamento che sono comparsi in queste pagine. Tenete presente che anche se vedete dal listato 1 che l'indirizzo di inizio da me scelto è stato 50000, la routine è completamente rilocabile, cioè potete caricarla all'indirizzo che preferite, senza dover modificare niente. Ovviamente l'argomento della USR nella DEF FN dovrà essere il valore scelto da voi.

Il listato 3 è invece un programma di prova, in cui voi date i due parametri e vi viene stampato in risposta il valore del tasto corrispondente. Provate anche con valori errati, per rendervi conto del meccanismo di errore. Unica limitazione del programmino, data dalla peraltro indispensabile presenza di PAUSE 0, è l'insensibilità alla pressione dei soli SHIFT; per provarli, premeteli insieme a qualche altro tasto.

Per quanto riguarda la meccanica interna della routine rimando all'Assembly, qua e là commentato, facendo solo notare come la lettura sia totalmente indipendente dallo stato degli altri bit e sia quindi indifferente alla versione di Spectrum usata.

COMMODORE VIC 20 e C 64

Magic Key

di Alessandro Guida

Abbiamo constatato che l'invito (tutt'ora valido) a scriverci dei suggerimenti, piccole routine e segreti scoperti dei due piccoli Commodore ha avuto notevole successo.

Mentre continuiamo a leggere attentamente le vostre lettere, siamo giunti ad una prima conclusione. Siete molto interessati a utility che rendano il più veloce possibile la stesura dei programmi.

Il 64 ed il VIC sono già dotati di un editor molto efficiente che permette, tra l'altro, di abbreviare i comandi in sole due o tre lettere. Certo, sarebbe comodo poter premere, addirittura, un solo tasto ed

I SEGRETI DEI PERSONAL

Magic Key

```
10 PRINT"␣ 64 MAGIC KEY"
20 FORI=52557TO52739
30 READA:X=X+A:POKEI,A
40 NEXTI
50 IFX<>24016THENPRINT"C'E' UN ERRORE NEI DATA !":END
60 PRINT "SYS 52557 PER ATTIVARE"
70 DATA 120,173,20,3,72,173,21,3,72,173,116,205,208,2,169
80 DATA 118,141,20,3,173,117,205,208,2,169,205,141,21,3,104
90 DATA 141,117,205,104,141,116,205,88,96,0,0,72,138,72,152
100 DATA72,165,215,72,165,212,240,4,104,76,221,205,104,201,193
110 DATA144,82,201,219,176,78,56,233,193,170,189,229,205,162,0
120 DATA134,198,170,160,158,132,34,160,160,132,35,160,0,10,240
130 DATA16,202,16,12,230,34,208,2,230,35,177,34,16,246,48
140 DATA241,200,177,34,48,17,8,142,255,205,230,198,166,198,157
150 DATA119,2,174,255,205,40,208,234,230,198,166,198,41,127,157
160 DATA119,2,169,20,141,119,2,230,198,104,168,104,170,104,76
170 DATA49,234,198,169,199,134,128,129,161,144,133,137,141,200,202
180 DATA130,159,151,194,201,196,163,183,197,131
190 DATA135,140,158,127,0,0,0,255,255,255
```

Listato 1. *Versione per C 64.*

A	ASC	N	NEXT
B	STEP	O	OPEN
C	CHR$	P	POKE
D	DIM	Q	PEEK
E	END	R	RIGHT$
F	FOR	S	STR$
G	GET	T	TAB(
H	STOP	U	USR
I	INPUT	V	VAL
J	GOTO	W	DATA
K	GOSUB	X	READ
L	LEFT$	Y	RESTORE
M	MID$	Z	SYS

Figura 1. *Parole chiave assegnate ai singoli tasti.*

avere subito la parola chiave.
Un primo tentativo è già stato fatto ed è apparso alcuni numeri fa su **Personal Software**. Esso utilizzava la possibilità di introdurre dei simboli grafici dopo una REM. Una volta eliminata la REM quei simboli venivano interpretati dal computer come parole chiave. Vi sono però tre inconvenienti. Il primo dovuto alla necessità di cancellare tutte le REM prima di dare il via al programma. Il secondo per il fatto che non c'è alcuna corrispondenza tra il tasto shiftato e la parola chiave relativa e il terzo, infine, è che sullo schermo rimangono dei simboli grafici al posto dei comandi BASIC, finché non viene chiesto un LIST.
Con il piccolo programma presentato è invece possibile fare in modo che, alla pressione di un tasto alfabetico insieme allo SHIFT, sullo schermo appaia una parola chiave per intero. Inoltre, la stessa routine si preoccupa di controllare se tale parola trova tra virgolette, nel qual caso appare normalmente il carattere grafico.
La routine è semplicissima da utilizzare. Basta digitarla (ce n'é una per il VIC e una per il 64) e dare il RUN. Se non ci sono errori nei DATA, apparirà la scritta:

SYS nnnnn PER ATTIVARE

Bisogna dare il SYS, riportato, per attivare la routine. Un secondo SYS allo stesso indirizzo (o la pressione dei tasti STOP + RESTORE) la disabilita.
La figura 1 riporta tutte le parole chiave assegnate ai vari tasti. Sono state scelte le parole di uso più comune e, dove possibile, assegnate ai tasti corrispondenti all'iniziale della parola stessa.
Dopo aver dato il RUN il programma in BASIC si può cancellare dalla memoria.

Come funziona

La routine è stata scritta in linguaggio macchina (il listato 3 è il disassemblato della versione per il 64) ed utilizza alcune particolarità dei 64/VIC 20.
1) Si è sfruttata la possibilità di intervenire sulla

I SEGRETI DEI PERSONAL

Magic Key

```
10 POKE55,77:POKE56,29
20 PRINT "⌐VIC MAGIC KEY"
30 FORI=7501TO7679
40 READA:X=X+A''    POKEI,A
45 NEXT
50 IFX<>22351THENPRINT"C'E' UN ERRORE NEI DATA!":END
60 PRINT"SYS7501 PER ATTIVARE"
70 DATA120,173,20,3,72,173,21,3,72,173,116,29,208,2,169,118
80 DATA141,20,3,173,117,29,208,2,169,29,141,21,3,104,141,117
90 DATA29,104,141,116,29,88,96,0,0,72,138,72,152,72,165,215
100 DATA72,165,212,240,4,104,76,221,29,104,201,193,144,82,201,219
110 DATA176,78,56,233,193,170,189,229,29,162,0,134,198,170,160,158
120 DATA132,34,160,192,132,35,160,0,10,240,16,202,16,12,230,34
130 DATA208,2,230,35,177,34,16,246,48,241,200,177,34,48,17,8
140 DATA142,255,29,230,198,166,198,157,119,2,174,255,29,40,208,234
150 DATA230,198,166,198,41,127,157,119,2,169,20,141,119,2,230,198
160 DATA104,168,104,170,104,76,191,234,198,169,199,134,128,129,161,144
170 DATA133,137,141,200,202,130,159,151,194,201,196,163,183,197,131,135
180 DATA140,158,127
```

Listato 2. *Versione per VIC 20.*

routine di interrupt per aggiungervi la nostra, che esegue il controllo del tasto premuto prima che questo venga riversato sullo schermo.

Per chi non lo sapesse, la routine di interrupt è quella che si occupa, anche, della gestione della tastiera (vedi **Personal Software** n. 15, 16).

La parte di programma compresa tra $CD4D e $CD73 modifica il contenuto del vettore che punta all'inizio della routine di IRQ.

2) La locazione $ D7 (215), interrogata in $ CD7B, contiene il codice ASCII dell'ultimo tasto premuto.

3) La locazione $ D4 (212), che compare all'indirizzo $CD7E, è un flag posto a zero se l'editor non si trova in modo virgolette (quello attivato aprendo le virgolette e che causa l'apparizione di simboli grafici al posto delle funzioni come, ad esempio, il CLR che fa apparire un cuore in campo inverso).

Se il flag è settato (editor in modo virgolette) la nostra routine viene terminata ($CD83) saltando alla routine di IRQ originale.

4) Viene utilizzata la tabella dell'interprete BASIC contenente tutte le parole chiave, locata in $A09E per il 64 e in $C09E per il VIC.

Un altro piccolo segreto

Molte volte i programmi contengono dei DATA che vengono letti e immessi direttamente in memoria con una POKE. Se, ad esempio, nel listato 1 (o 2) sostituite uno dei dati con un 256, avrete un ILLEGAL QUANTITY ERROR IN 30 (o 40), poiché non è posibile utilizzare il POKE con numeri superiori a 255. Il problema è che il computer non vi dice in che linea vi è il DATA errato. Per lo meno non ve lo dice apertamente.

Infatti, le locazioni $3F (63) e $40 (64) contengono sempre il numero dell'ultima linea di DATA letta.

È perciò sufficiente digitare in modo diretto:

PRINT PEEK (63) + PEEK (64) ★ 256

e si avrà il numero della linea difettosa. ■

In redazione continuiamo ad attendere suggerimenti da tutti gli utenti Commodore.

Se avete scoperto qualche piccolo trucco, se avete suggerimenti da darci, o se avete idee da sviluppare e programmi completi, inviateli a:

Alessandro Guida
c/o **Personal Software** - Via Rosellini, 12 - 20124 Milano.

Listato 3. *Disassemblato della routine proposta.*

```
CD4D 78         SEI
CD4E AD 14 03   LDA $0314
CD51 48         PHA
CD52 AD 15 03   LDA $0315
CD55 48         PHA
CD56 AD 74 CD   LDA $CD74
CD59 D0 02      BNE $CD5D
```

Magic Key

Seguito listato 3.

```
CD5B A9 76       LDA #$76
CD5D 8D 14 03    STA $0314
CD60 AD 75 CD    LDA $CD75
CD63 D0 02       BNE $CD67
CD65 A9 CD       LDA #$CD
CD67 8D 15 03    STA $0315
CD6A 68          PLA
CD6B 8D 75 CD    STA $CD75
CD6E 68          PLA
CD6F 8D 74 CD    STA $CD74
CD72 58          CLI
CD73 60          RTS
CD74 00          BRK
CD75 00          BRK
CD76 48          PHA
CD77 8A          TXA
CD78 48          PHA
CD79 98          TYA
CD7A 48          PHA
CD7B A5 D7       LDA $D7
CD7D 48          PHA
CD7E A5 D4       LDA $D4
CD80 F0 04       BEQ $CD86
CD82 68          PLA
CD83 4C DD CD    JMP $CDDD
CD86 68          PLA
CD87 C9 C1       CMP #$C1
CD89 90 52       BCC $CDDD
CD8B C9 DB       CMP #$DB
CD8D B0 4E       BCS $CDDD
CD8F 38          SEC
CD90 E9 C1       SBC #$C1
CD92 AA          TAX
CD93 BD E5 CD    LDA $CDE5,X
CD96 A2 00       LDX #$00
CD98 86 C6       STX $C6
CD9A AA          TAX
CD9B A0 9E       LDY #$9E
CD9D 84 22       STY $22
CD9F A0 A0       LDY #$A0
CDA1 84 23       STY $23
CDA3 A0 00       LDY #$00
CDA5 0A          ASL
CDA6 F0 10       BEQ $CDB8
CDA8 CA          DEX
CDA9 10 0C       BPL $CDB7
CDAB E6 22       INC $22
CDAD D0 02       BNE $CDB1
CDAF E6 23       INC $23
CDB1 B1 22       LDA ($22),Y
```

Seguito listato 3.

```
CDB3 10 F6       BPL $CDAB
CDB5 30 F1       BMI $CDA8
CDB7 C8          INY
CDB8 B1 22       LDA ($22),Y
CDBA 30 11       BMI $CDCD
CDBC 08          PHP
CDBD 8E FF CD    STX $CDFF
CDC0 E6 C6       INC $C6
CDC2 A6 C6       LDX $C6
CDC4 9D 77 02    STA $0277,X
CDC7 AE FF CD    LDX $CDFF
CDCA 28          PLP
CDCB D0 EA       BNE $CDB7
CDCD E6 C6       INC $C6
CDCF A6 C6       LDX $C6
CDD1 29 7F       AND #$7F
CDD3 9D 77 02    STA $0277,X
CDD6 A9 14       LDA #$14
CDD8 8D 77 02    STA $0277
CDDB E6 C6       INC $C6
CDDD 68          PLA
CDDE A8          TAY
CDDF 68          PLA
CDE0 AA          TAX
CDE1 68          PLA
CDE2 4C 31 EA    JMP $EA31
CDE5 C6 A9       DEC $A9
CDE7 C7          ???
CDE8 86 80       STX $80
CDEA 81 A1       STA ($A1,X)
CDEC 90 85       BCC $CD73
CDEE 89          ???
CDEF 8D C8 CA    STA $CAC8
CDF2 82          ???
CDF3 9F          ???
CDF4 97          ???
CDF5 C2          ???
CDF6 C9 C4       CMP #$C4
CDF8 A3          ???
CDF9 B7          ???
CDFA C5 83       CMP $83
CDFC 87          ???
CDFD 8C 9E 7F    STY $7F9E
CE00 00          BRK
CE01 00          BRK
CE02 00          BRK
CE03 FF          ???
CE04 FF          ???
CE05 FF          ???
.
.
```

PICCOLI ANNUNCI

Apple

Scambio software Apple II e cerco fotocopie del programma Space Shuttle, che posso pagare con un programma.
Fabrizio Vettore - Via S. Croce 327 - 30125 Venezia - Tel. 041/700866.

Cambio programmi di qualsiasi tipo per simulazioni e games molto complessi (tipo time zone, marauder, southern command...) **per Apple II**. Dispongo dBase 2, the last one, Tasc, Cobol...
Stefano Malagodi - Via Comunale 14 - 44034 Copparo (FE) - Tel. 0532/860196.

Cambio programmi di ogni genere per **Apple**. Possiedo 130 gestionali e 100 giochi. Speditemi vostra lista.
Tiziano Maffei - Il Podere Loc. S. Giovanni - 57037 Portoferraio (LI) - Tel. 0565/916566.

Cambio/vendo software per **Apple II**, scientifici, gestionali, utilità, giochi, vasta scelta. Inviare la lista a cui risponderò con la mia.
Fabrizio Vettore - Via S. Croce 327 - 30125 Venezia - Tel. 041/700866.

Software per **Apple cambio e vendo** contabilità generale clienti, fornitori, fatture, stipendi, ristoranti, alberghi, condomini, numerose utility, ingegneria e gestione testi, visicorp software.
Marco Nundini - Via E. Oddone 17 Rivalta - 42100 Reggio Emilia - Tel. 0522/569219.

Apple tastierino programmabile e silentype con interfaccia **vendesi**. Telefonare ore 13-14.
Fabio Dimant - Via F. Raibolini 33/7 - 60069 Zola Predosa (BO) - Tel. 051/273277.

Cerco per **Apple II: Mumathimusimp**, logica simbolica, fisica, ottica, micro expert, prolog, diagnosis, mathemagic/graphmagic combo, programmi e linguaggi per l'intelligenza artificiale.
Antonio Scala - Via G. Imbroda 39 - 80035 Nola (NA) - Tel. 081/8234710.

Vendo/cambio **programmi e schede per Apple II IIe**, vasta disponibilità a prezzi stracciati. Scrivere per informazioni.
Carlo Casalicchio - Via M. Velino (Residence Ameno) - 60100 Ancona.

Scambio con utenti **Apple programmi** di ogni tipo. In particolare giochi, utility, didattica. Compro anche programmi per copiare i floppy. Telefonatemi o venitemi a trovare.
Massimo Gilardetti - Via Borsi 62 - 28100 Novara - Tel. 0321/474639.

Cambio programmi per **Apple II**. Sono in possesso di : Dechatlon, guerre spaziali (e giochi inclusi), guerra mondiale att. Tutti su disco. Chiamare ore pasti.
Simone Serafini - P.zza Ara Coeli 6 - 00186 Roma - Tel. 6790146.

Cambio/vendo programmi applicativi, utility e giochi per **Apple II**. Inviare L. 500 o spedire la propria lista. Risponderò con l'elenco dei miei programmi.
Paolo Vercellesi - Via M. Pichi 11 - 20143 Milano - Tel. 02/8394831.

Scambio programmi per **Apple II**. Mandatemi vostra lista ed io vi manderò la mia, contenente 250 programmi.
Tiziano Maffei - Il Podere Loc. S. Giovanni - 57037 Portoferraio (LI) - Tel. 0565/916566.

Vendo moltissimi **programmi** per **Apple** di tutti i tipi. Per avere una lista dei programmi telefonare al 3275666 di Roma, al pomeriggio verso le ore 6.30, e chiedere di Karim.

Esperto **programmatore CNR** cambia **programmi** per **Apple** solo su dischetto.
Alberto Pollastri - Via Di Pratale 28B - 56100 Pisa - Tel. 050/20584.

Cambio programmi per **Apple**, dispongo di una vasta biblioteca di software. Disposto a scambiarli anche con hardware. Rispondo a chiunque mi contatta anche per scambio esperienze su Apple.
Franco Vittor - Via Grabizio 35 - 34170 Gorizia - Tel. 0481/81254 ore serali.

Vendo per **Apple II IIe programmi** di ogni genere + manuali a prezzi bassissimi; molte novità. Telefonare ad Andrea 06/3201782.

Vendo listati programmi **Apple e programmi** per **Vic 20** a basso prezzo.
Giovanni Bellamacina - Via Mazzini 31 - 20058 Villasanta (MI) - Tel. 039/302576.

Cambio programmi per **Apple IIe** e per Commodore 64. Inviate la vostra lista, io vi invierò la mia.
Giovanni Cutolo - Via Tasso 97 - 80126 Napoli.

Scambio per **Apple II** 250 **programmi** e vendo 1 drive appena provato con garanzia.
Tiziano Maffei - Il Podere Loc. S.Giovanni - 57037 Portoferraio (LI) - Tel. 0565/916566.

Vendo per **Apple II**: "Language card" originale (16K espansione) a L. 155.000, prezzo di listino L. 352.000 + IVA. Inoltre compro programma per "Epron Programmer" di **Bit** Febb. 83.
Tiziano Settimi - Via XXIV Maggio 30 - 20010 Canecirate (MI) - Tel. 0331/400303.

Apple IIe ancora imballato, fattura Febbraio 84, scheda 80 colonne, espansione memoria 128K, 2 Disk Driver, Monitor Apple, stampante ad aghi, interfaccia parallela, tutto **40% di sconto**. Telefonare ore 20-23.
Gabriella Bellini - Via C. De Cesare - 80100 Napoli - Tel. 081/661781.

Compro a prezzo d'occasione **Drive 5"** per **Apple** compatibili. Telefonare ore serali allo 02/5270019 e chiedere di Fabrizio.

Cerco possessori **Apple** per scambio di esperienze e programmi.
Melchiorre Schifano - Via Padova 10 - 10152 Torino Tel. 011/270324.

Vendo in blocco **programmi** del totocalcio su **Apple II** : corr. err., sistema chiave 6 Alfa super, sistema a rotazione, sistemi ridotti, analisi di campionato + altri tipi di sistema a L. 100.000 (compresa spedizione + disco).
Telefonare allo 085/207466.

Apple II IIe a prezzi competitivi, **programmi** di ogni genere : utility, giochi, gestionali, grafica, ecc. Ulteriori sconti saranno praticati in ragione del numero di programmi acquistati.
Luigi Palumbo - Via Adelaide Ristori 8 - 00197 Roma - Tel. 06/802783.

Vendo o **scambio** compatibili **Apple** molti **programmi** applicativi e giochi. Ad interessati invio elenco. Unire alla richiesta L. 1.000 in francobolli per rimborso spese.
Lanza Mario - Via Teano 21. - 20161 Milano - Tel. 02/6453879.

Vendo per Apple II "Supertoto 1.0"", superprogramma inedito totocalcio, 3 diverse opzioni di selezioni incrociate; con output n° colonne utili, sviluppo su Monitor o stampante, a L. 70.000 con man.
Roberto Rossi - Via Lario 26 - 20159 Milano - Tel. 02/6070236.

Commodore

Cerco assolutamente **possessore del Tool's kit**. Il mio segnala errore in 500. Organizzazione nuovo club "Commodore"; ho bisogno di moltissimo materiale, rimborserò qualsiasi spesa postale. Grazie a quanti risponderanno.
Fabrizio Aureli - Via Delle Grazie 96 - 60100 Ancona - Tel. 071/84580.

Vendo/compro/scambio **programmi per Commodore 64**, invio lista a chi mi invia la propria, oppure L. 700 in francobolli.
Lauro Michelotti - Via Boboli 1/A - 51017 Pescia (PT) - Tel. 0572/477727.

Vendo 30 fantastici **programmi** per **VIC 20**, come totocalcio, life, bioritmi, invaders, drawer, ecc. Su cassetta in blocco L. 29.000 tutti.
Mazza Armando - Via Settembrini 96 - 70053 Canosa (BA) - Tel. 0883/64050.

Eccezionale si è costituito a Torino un **Commodore VIC 20 club** a livello nazionale. Chi ha intenzione di aderirvi invii L. 2.000 precisando nome, cognome e indirizzo. Gli saranno mandate informazioni.
Maurizio Moretti - C.so Monte Cucco 33 - 10139 Torino - Tel. 011/3359684.

Cambio/compro/vendo **programmi** per **VIC 20**. Cerco Super Expander a buon prezzo.
Leonardo Pecchi - Via SS per Correggio 22 - 41012 Carpi (MO) - Tel. 059/664219.

Vendo registratore per **VIC** e **64**; Superexpander + 3K RAM, cartridge Jelly Monsters, per VIC 20, cartridge Avenger per 64, tutto come nuovo.
Paolo Bondino - Via Montanara 2 - 05100 Terni Tel. 0744/82170.

Scambio circa 500 **programmi** per **Commodore 64**. Richiedere lista a:
Paolo Raimondo - C.so Vittorio Emanuele 71 - 10128 Torino - Tel. 011/545625.

Cerco joystick per C64.
Andrea Giannotti - Via Milano 16 - 55042 Forte dei Marmi (LU) - Tel. 0584/82302.

Per **CBM 64 compro programmi** di qualunque tipo. Cerco procedure di back up cassetta-cassetta, anche interfaccia. Inviare liste prezzi a:
Leonardo Lombardi - Largo A. Banfi 4 - 50018 Scandicci (FI) - Tel. 055/256640.

Cambio e **vendo programmi** per **VIC 20**. Massima serietà e validissimi programmi. Contattatemi.
Marco Ceruti - Via Frassinago 59 - 40132 Bologna - Tel. 051/585384.

PICCOLI ANNUNCI

Sono interessato a contattare **club Commodore 64** per adesione oppure scambio vendita acquisto software.
Marco Bombonato - Via Nino Bixio 27 - 20129 Milano.

Per **C64** cavo di interfaccia parallela Centronics da inserire nella User-port + routine di abilitazione in L.M.. Si usa da BASIC con un print e manda dati a qualunque unità periferica Centronics compatibile. L. 59.000 + ss. contrassegno.
Lorenzo Fontani - Via Cavour 33 - 53100 Siena - Tel. 0577/284229.

Vendo/cambio per **VIC 20** centinaia di programmi su cassetta a prezzi strabilianti anche in LM.
Antonio Luini - Via Parco dei Principi 50 - 70010 Casamassima (BA).

Vendo per C64 software originale ricreativo e didattico. I programmi vengono forniti su cassetta. Invio catalogo gratuitamente. Produco software gestionale su misura anche per Apple IIe.
Sergio Fumich - Via Togliatti 1/A - 20070 Brembio (MI).

Cerco programmi pronti (giochi) su cartridge per **CBM 64** a buon prezzo. Annuncio sempre valido. Telefonate, cerco inoltre utenti zona Milano per scambio idee e software. Telefonare ore pasti.
Fabrizio Boti - Via Papa Giovanni XXIII 23 - 20093 Cologno Monzese (MI) - Tel 02/2535812.

Vendo, per passaggio a sistema superiore, **VIC 20** ancora in garanzia insieme a, pensate, ben 230 programmi su cassette (giochi, utility, matem.) e 1 cartuccia gioco. Tutto a L. 200.000.
Domenico Basile - Via A. Volta 13 - 74012 Crispiano (TA) - Tel. 099/616556.

Vendo e/o scambio **giochi per VIC 20**, posseggo più di 350 programmi. Il tutto a prezzi bassissimi, vendesi in blocco 250 giochi per VIC 20 (tra cui Bonzo, Sub chase, Arcadia, Frogger). Il tutto a L. 50.000
Alessandro Natali - Viale Abruzzi 28 - 20131 Milano - Tel. 02/2713636.

Cambio/vendo moltissimo **software** per **CBM 64**. Linguaggi, compilatori, gestionali, utilities, videogames e molti altri. Telefonare ore 14-15.
Paolo Vergoni - Via Appia 10 - 06100 Perugia - Tel. 075/66918.

Cerco qualsiasi tipo di **programma** per **CBM 64** a prezzo non eccessivo. Bastano anche le fotocopie dei listati.
Nicola Pisilli - Via Molinetto di Lorenteggio 15 - 20094 Corsico (MI) - Tel. 4408947.

Vendo, causa passaggio a sistema superiore, **C64 + unità nastro** + 2 joystick 1311 + hesmon64 con trad. it. + 2 giochi su cassetta + decine di programmi a L. 850.000 trattabilissime.
Renato Fornasero - Viale Trieste 281 - 93100 Caltanissetta.

CBM 64 scambio **programmi** e vendo. Cerco programmi per gestione stazioni radioamatoriali.
Vincenzo Petrizzo - Via Nazionale 157 - 84030 Padula Scalo (SA) - Tel. 0975/74022.

Vendo per **VIC 20** bellissimi **videogiochi** ad un prezzo interessante. Risponderò a tutti. Meglio se telefonate.
Luca Cioffi - Via Adua 7 - 44100 Ferrara - Tel. 0532/48982.

Vendo programmi per **VIC 20** a prezzi veramente incredibili. Scrivere per maggiori informazioni.
Massimo Tabasso - Piazza Molineris 1 - 12038 Savigliano (CN).

Vendo/scambio splendidi **videogames e programmi** utility su nastro o su disco **per C64**. Scrivetemi: oltre al catalogo dettagliato riceverete un gioco in omaggio e tantissimi altri regali.
Guido Pellicani - Via Devito Francesco 27 - 70124 Bari - Tel. 080/221015.

Vendo per CBM 64 giochi interessanti. Vendo inoltre utilities come gestione c/c, word italiano e il Simon's BASIC con manuale. Scrivere o telefonare ore pasti a:
Francesco Punzo - Via Tufarelli 18 - 80046 San Giorgio a Cremano (NA) - Tel. 081/7716827.

Per C64 vendo/scambio **software** a basso costo; in particolare: Simon's BASIC- dimostrazioni e manuale istruzioni; NEWBASIC; Turbo (riduce la lunghezza di un programma su nastro di 1/10 di giri); invio catalogo anche di giochi; possibilmente spedire il francobollo per mia risposta.
Alfonso Dinoia - Via Arrigo Davila 61 - 00179 Roma - Tel. 06/7884472.

Compro/cambio/vendo **software** per **CBM 64**. Dispongo di una notevole quantità di giochi su cassetta e disco in LM. Per ricevere la lista dei giochi allegare L. 800 o telefonare.
Giammaria Lari - Via Sinigaglia 3 - 26013 Crema - Tel. 0373/81117.

Sinclair

Vendo per **ZX Spectrum** 16K cassetta contenente moltissimi **programmi**, giochi di ogni genere, analitica. L. 12.000 + spese postali. Massima serietà.
Francesco Landogna - Via Cavour 110 - 96017 Noto (SR).

Vendo ZX Spectrum 48K + registratore Philips Data Recorder + manuale in Italiano + 250 programmi originali su cassetta. Il tutto a L. 600.000 tratt.
Francesco Ippolito - Via Cales 4 - 00050 Cerenova (Roma) - Tel. 06/9931157.

Cambio **programmi** per **ZX Spectrum**. Oltre 350 titoli a disposizione. Inviare propria lista e L. 500 per ricevere dettagliate istruzioni. Scrivere se avete almeno 80 titoli.
Luigi Ballestrin - Via Martiri della Libertà 367/11 - 18038 Sanremo (IM).

Vendo per **ZX Spectrum** circa 300 **programmi** in blocco a L. 120.000. Tutti i migliori e le ultime novità. Spedizione contrassegno se fuori zona dietro anticipo di L. 10.000.
Marino Marinanza - Via Rastrelli 102 - 00128 Roma - Tel. 06/5203292.

Vendo programmi per **ZX Spectrum** a L. 2.000 per 16K e L. 4.000 per 48K. Possibilità di scambio. Spedire L. 400 in francobolli per lista software. Telefonare fuori orario pasti.
Luca Bresin - Tel. 041/711141 - Venezia.

ZX Spectrum. Il **Sinclair Club di Mantova** dispone di oltre 700 programmi in continuo aggiornamento. Siamo a vostra disposizione per qualsiasi chiarimento ed informazioni.
Luca Giambellini - Strada Cisa 95 - 46047 Bamcole (Mantova) - Tel. 398220.

Per **Spectrum** 48K **cedo** (causa incompetenza) **cassette** di Forth e Pascal con manuale in Italiano a L. 10.000 ciascuna + spese postali.
Ennio Metta - Via A. Gioia 122 - 70054 Giovinazzo (BA).

Vendo/cambio moltissimi **programmi** per **ZX Spectrum** 16 e 48K a prezzi veramente bassi. Scrivere o telefonare per elenco completo a:
Piero Cingolani - Via Dei Pini 37 - 62017 Porto Recanati (MC) - Tel. 071/9798853.

Cambio **programmi** di ogni tipo per lo **ZX Spectrum**. Inviate la vostra lista, risponderò con la mia: risposta assicurata. Per maggiori informazioni scrivere o telefonare a:
Igor Bonat - Via Ressmann 6 - 34141 Trieste - Tel. 040/768718.

Cambio 200 **programmi** per **ZX Spectrum**. Tutti originali inglesi, LM e con istruzioni. Ultime novità. Rispondo a tutti. Massima serietà.
Vincenzo Tomassi - Via G. Leopardi 34A - 03043 Cassino (FR) - Tel. 0776/24109.

Cambio per **ZX Spectrum** circa 40 **programmi** su cassetta (giochi, utilità, gestionali). Inviatemi la vostra lista, io farò altrettanto. Contattatemi anche telefonicamente (ore pasti).
Fabio D'Urso - P.za S. Giuseppe 10 - 20162 Milano - Tel. 02/6433677.

Vendo e cambio **programmi** per **ZX Spectrum** a prezzi incredibili. Tantissime novità ogni settimana. Massima serietà e convenienza. Telefonare dalle 14 alle 19.30 o scrivere a:
Alessandro Carbonara - Via Faenza 159 - 70019 Triggiano (BA) - Tel. 080/681928.

Compro Sinclair Z 80 ancora funzionanti, oppure quelli guasti. Sono disposto a pagare tutte le spese postali. Per un contatto telefonare o scrivere a:
Andrea Meroi - Via Mignone 11 - 52100 Arezzo - Tel. 0575/29091.

Per **ZX 81** scambio **programmi** utili e progetti hardware. Inviare L. 400 in francobolli specificando cosa interessa. Possiedo materiale anche per i più comuni computer.
Fabrizio Martano - Via Don L. Sturzo 7 - 58100 Grosseto - Tel. 0564/492806.

Vendo per **Spectrum** Eccezionali **giochi** in linguaggio macchina. Cassetta 10 giochi 16K a L. 10.000. Cassetta 10 giochi 48K a L. 15.000. Pagamento in contrassegno L. 2.000 per spese postali.
Fabio D'Amico - Via 5 Giornate 14 - 20025 Legnano (MI) - Tel. 0331/595581.

Vendo per **Spectrum** 16/48K **programma** su cassetta che risolve qualunque triangolo a L. 15.000, comprese le spese di spedizione. Inviare vaglia postale o assegno bancario.
Anacleto Furlan - Viale Papa Giovanni 35/10 - 31015 Conegliano (TV).

PICCOLI ANNUNCI

Acquisto espansione di **memoria** per **ZX 81**, inoltre compro giochi solo se veramente belli. Solo Milano. Telefonare ore pasti.
Luca Mantegazza - Via Mosè Bianchi 26 - 20149 Milano - Tel. 02/431605.

Ora c'è il **"Sinclub Napoli Ovest"**. Centinaia di programmi per Spectrum 16/48K. Scambio gratuito e iscrizione gratuita. Per informazioni e lista programmi telefonare nel pomeriggio.
Marco Antignano - Via Cupa S. Cesarea 33 - 80145 Napoli - Tel. 081/7553580.

Cambio **programmi ZX Spectrum** 16/48K. Inviate la vostra lista per ricevere, oltre alla mia, alcuni programmi che non possedete, in attesa di ricevere i vostri. Rispondo a tutti immediatamente.
Marco Ghirardelli - Via Caduti per la Libertà 30 - 48015 Cervia (RA) - Tel. 0544/971697.

Cerco software per **Spectrum** 48K riprodotto su cassetta vergine o su cassetta originale.
Cristiano Magistrali - Via Cristalli 5 - 29100 Piacenza - Tel. 0523/384807.

Spectrum, vendo **programma** totocalcio toto PM V2, sistemi integrali, statistici, condizionati, base ibrida, sezioni, schede multiple. Per dettagli e informazioni spedire L. 400 in francobolli.
Mario Pacchiarotti - Via Tuscolana 884 - 00174 Roma (RM) - Tel. 06/7612788.

Scambio/vendo **programmi** ogni genere per **Spectrum** 16/48K. Risposta rapida e certa. Inviatemi la vostra lista ed entro pochi giorni riceverete la mia.
Enzo Bon - Via Domenico Loschi 11/A - 31100 Treviso (TV) - Tel. 0422/66298.

Cerco, zona Cecina/Rosignano (LI), **possessori ZX Spectrum** per scambio idee ed, eventualmente, fondazione club.
Alessandro Paroli - Via 2 Giugno n. 2 - 57020 La California (LI) Tel. 0586/677207.

Cerco **Spectrum** 48K (tratto anche per il 16K), in cambio offro televisore a colori Brionvega 24 pollici (ha circa 10 anni ed è leggermente starato) funzionante. Tratto con Milano e provincia.
Fabio Confalonieri - Via Sicilia 13 - 20038 Seregno (MI) - Tel. 0362/220820.

Compro, vendo e scambio **software** per **ZX Spectrum**. Ultimi arrivi dall'Inghilterra. Prezzi ottimi e forti sconti per grossi quantitativi. Disponibili oltre 200 programmi.
Mauro Costa - Via Monte Suello 2/12A - 16129 Genova - Tel. 010/314704.

Vendo a prezzi ridicoli **programmi** per **ZX Spectrum**. I programmi (16 e 48K) sono registrati direttamente da computer. E veramente un'occasione unica.
Alberto Pilesi - Via D' Ancona 32/B - 60027 Osimo (AN).

Vendo Sinclair ZX81 imballo originale usato pochissimo. 16K Ram + cavi, manuali vari e cassette programmi. L. 250.000 tratt.
Carlo Marazzini - C.so Lodi 50 - 20139 Milano - Tel. 02/5691858.

Il **"Sinclair Club Mantova"** annuncia la sua costituzione mettendosi a disposizione di chi volesse cambiare o vendere Software per Spectrum 16 o 48K. Vi aspettiamo C.M.
Luca Gandellini - Strada Cisa 95 - 46047 Mantova - Tel. 0376/398220.

Programmi a prezzi eccezionali per **Spectrum**, oltre 150 titoli, tutti originali. Una cassetta con 20 titoli a vostra scelta L. 10.000.
Richiedere lista gratuita. Massima serietà.
Massimo Carosi - Via del forte Tiburtino Pal.18 sc.C - 00100 Roma.

Vendo a prezzi stracciati più di 200 **programmi** per **Spectrum**, tra cui le ultime novità (Wahalla, Atic Atac Lunar Jetman...).
Marco Bartoli - Via Pinturicchio 10 - 05100 Terni - Tel. 0744/420282.

Vendo/scambio **software** per **ZX Spectrum** 16/48K come Manic-Miner, Psst, Zzum, e vari programmi di utilità. Inviare listino.
Marco Gianferrara c/o Bottarelli - Via G. Agnelli 6 - 56100 Pisa.

Cerco possessori **ZX Spectrum** per scambio programmi. Possiedo circa 150 programmi per Spectrum 16/48 Kbyte.
Stefano Lazzarino - Via G. Arganini 22 - 20162 Milano - Tel. 02/6437645.

Acquisto libri per **ZX Spectrum** (in italiano). Cerco **ZX Printer** (o simile) a prezzo modico. Scambio vendo acquisto programmi (Utility-Games) per ZX Spectrum.
Marco Arzani - Via Giovanni XXIII - 29017 Fiorenzuola D' Arda (PC) - Tel. 0523/9824401.

Causa passaggio a sistema superiore vendo **Spectrum 16K** ancora in garanzia 2 mesi di vita con 2 cassette gioco, programmi vari, manuale in italiano e libro programmi a L. 300.000.
Stefano Rigoldi - Via Visconti 30 - 20093 Cologno Monzese (MI) - Tel. 02/2547348.

Cerco programma di qualsiasi genere per computer **Sinclair ZX81** e **Spectrum**.
Per maggiori informazioni scrivere a:
Massimo Tabasso - P.zza Molineris 1 - 12038 Savigliano (CN).

Scambio **programmi** per **ZX Spectrum** 16K.
Antonio Santone - Via Fonte Regina 28 - 64100 Teramo - Tel. 0861/50464.

Vendo Sinclair ZX Spectrum 48K RAM con coperchio esterno di uno ZX nuovo. Lo Spectrum, è un issue two ed è in vendita dotato di 20 programmi, alcuni originali inglesi ed altri con grafica superiore. Cedo tutto a L. 500.000 (prezzo d'aquisto L. 950.000). Scambio inoltre **Software** per **Spectrum**.
Luca Raggi - Via Caccialupo 4 - 29100 Piacenza - Tel. 0523/21163.

Spectrum 48K ti sfida a Push Over, gioco di strategia su scacchiera.
Spectrum può anche solo arbitrare. Programma su cassetta a L. 20.000 compresa spedizione. Inviare assegno o vaglia postale.
Anacleto Furlan - V.le Papa Giovanni 35/10 - 31015 Conegliano (TV).

Spectrum- cambio/vendo **programmi** gioco e/o utilità rapida. Risposta elenco gratuito. Scrivere a:
Antonio Sfriso - Via O. Salomone 7 - 30173 Mestre (VE) - Tel. 041/972887.

Vendo per **ZX Spectrum e ZX81** sintesi vocale (con manuale) a L. 100.000, vendo, inoltre, ZX Printer comprata 2 mesi fà a L. 120.000.
Paolo Agnelli - Via Papa Giovanni XXIII 43 - 20091 Bresso (MI) - Tel. 02/6102190.

Vendo **programmi** per **ZX Spectrum** Traxx, Lunar Jetman, Castle of the Doom, Road racer, MR Wimpy, Splat, Game Designer e molti altri. Catalogo a richiesta L. 1000.
Stefano Nobili - Via Tuscolana 224 - 00181 Roma.

Cerco possessori **ZX Spectrum** residenti a Monza per scambio Programmi.
Stefano Raffaele - Via Crescitelli 6 - 20052 Monza (MI) - 039/366372.

Cambio o eventulmente vendo **software** per **ZX Spectrum** tra i quali: tutti i giochi ultimate-software, penna ottica, VV3D, Manic miner e altri.
Vincenzo Arnone - Via R. Grieco 37 - 71100 Foggia.

Cerco **stampante (ZX Printer o altra)** per ZX Spectrum. Cedo in cambio oltre 100 programmi per ZX Spectrum 16/48K. Richiedetemi la lista completa.
Renato Cozzolino - Via P. Federico 22 - 80045 Pompei (NA).

Cambio/vendo numerosi **programmi** su cassetta per **ZX Spectrum**. Dispongo di oltre 140 novità a prezzi davvero incredibili. Gradisco anche scambio di idee e di listati.
Pasquale Di Tommaso - Via F. Rossi 10 - 70053 Canosa di Puglia.

Vendo 25 **programmi** per **ZX Spectrum** su cassetta (tutti inediti) a sole L. 5000 non perdete questa occasione. Spedizione contrassegno.
Sandro Romano - Via Adelasia 11 - 07046 Porto Torres (SS) - 079/514501.

Sono un ragazzo di 10 anni possessore di uno **Spectrum** 16K a cui servono tanti programmi giochi per divertirsi e imparare. Chi ne avesse in esubero scrivere o telefonare a:
Andrea Bosotti - Via Risorgimento 47 - 20098 S. Giuliano Mil. (MI) - Tel. 02/9843569.

Causa passaggio a sistema superiore **vendo ZX81 Sinclair** con garanzia non datata manuale istruzioni in italiano libro "66 programmi per ZX81" + programmi vari tutto a L. 100.000 o comunque al miglior offerente.
Diego Massioli - Via Varese 3 - 20030 Cassina Savina (MI) - Tel. 0362/509811.

Cambio o vendo **programmi**, giochi o utilità per **Sinclair** 16/48K (preferibilmente provincia di VE).
Luca Prian - Via Cannaregio 1091/I - 30121 Venezia - Tel. 713438.

Vendo 100 **programmi** in blocco per lo **Spectrum** a prezzi favolosi (L. 1000 cad.).
Michele Romano - Via A. Vivaldi 20 - 04100 Latina - Tel. 241315.

Offro causa doppione pratica **consolle** metallo ancora imballata, per più comodo uso e lettura funzioni **Spectrum** a L. 15.000
Riccardo Batti - V.le Umbria 36 - 20135 Milano - Tel. 02/5469734.

Scambio oltre 400 **programmi** per **Spectrum**.
Monica Mauro - C.so Montecucco 64 - 10100 Torino - Tel. 613714.

PICCOLI ANNUNCI

Vendo scambio acquisto **programmi** per **ZX Spectrum** 16/48K. Garantisco massima serietà, dispongo di più di 200 titoli.
Liviano Vitali - Via Firenze 29/3 - 41035 Massa Finalese (MO) - Tel. 0535 99700.

Spectrum cambio/vendo **software** e giochi originali inglesi con altri possessori Spectrum di Verona e provincia.
Niccolò Arrigoni - Lungadige Panvinio 35 - 37121 VR - Tel. 045/23583.

Vendo ZX81 + **ZX Printer** a L. 250.000. Inoltre TI994A + registratore Sanyo + Joystick + sss: chess, invader , othello, personal record keeping, video games 1 tutto a L. 450.000.
Massimo Serio - Via Campanile 2 tr pr. 21/B - 80126 NA - Tel. 081/7267438.

Vendo o cambio più di 150 **programmi** originali inglesi per **ZX Spectrum** 16/48K.
prezzo medio 16K L. 5000, 48K L. 8000. Sconti per grossi acquisti. Inviare proprie liste per scambi.
Paolo Siamonetta - Via A. Colombo 30 - 21012 Cassano Magnago (VA) - Tel. 0331/201927.

Vendo **programmi** per **ZX81** su listato a partire da L.2000. Inoltre scambio programmi per VIC 20. Chi fosse interessato telefoni (ore pasti) a:
Marco Nascè -Via Volturno 3 - 30174 Mestre (VE) - Tel. 980968.

ZX Spectrum 16/48K, vendo cassette (TDK) registrate: con 4 giochi a scelta L. 15.000, con 2 giochi a scelta L. 10.000. Inviare francobollo per catalogo.
Giorgio Favaro - Via Lamarmora 39 - 10093 Collegno (TO) - Tel. 011/7800974.

Vendo circa 200 **programmi** per **Spectrum** a prezzi bassi. Molte novità inglesi. Elenco gratis a richiesta.
Stefano calcaterra - Via Marconi 34/2 - 40122 Bologna - Tel. 051/521063.

Spectrum software utilizzabile didatticamente cerco e scambio scrivere per accordi.
Maurizio Monaldi - Via V.Montiglio 7 - 00168 Roma - Tel. 06/6283901.

Vendo **cassette** per **ZX Spectrum**. Costo da L. 5000 a L.10.000 (circa).
Ogni cassetta contiene minimo 7 programmi tra giochi e utilities. Telefonare o scrivere a:
Massimo De Lorenzis - Via monti Tiburtini 510 - 00157 Roma - Tel. 06/4505925.

Per **Spectrum Sinclair** vendo a sole L. 10.000 cassette con 6 programmi a scelta duplicati direttamente da computer e provati. Richiedere elenco gratuito scrivendo o telefonando a:
Daviae Di Dio - Via C. Alberto 46/a - 80045 Pompei (NA) - Tel. 081/8635055.

Spectrum software. A chi è in grado di fornirmi, su nastro, i programmi di questa rivista (od altre) offro del software a scelta tra quello disponibile nel mio elenco. Scrivere per accordi.
Maurizio Monaldi - Via V. Montiglio 7 - 00168 Roma - 06/6283901.

Tutti i migliori programmi **Spectrum**, libri, bollettino, adesivi, corsi L/M consulenza e tanti amici. Questo è il "gruppo utilizzatori computer Sinclair". Scriveteci allegando bollo, diventeremo amici.
Roberto Chimenti - Via L. Rizzo 18 - 80124 Napoli - Tel. 081/617368.

Scambio, alla pari, programmi 16/48K per **ZX Spectrum**. Disposto anche a comperare mandatemi vostra lista.
Francesco Rusca - Via Volturno 14 - 43100 Parma - Tel. 0521/54600.

Cambio programmi di ogni tipo per lo **ZX Spectrum**. Inviare la vostra lista-risposta assicurata. Risponderò con la mia lista: più di 300 titoli.
Igor Bonat - Via Ressman 6 - 43141 Trieste - Tel. 040/768718.

Computer ZX81 con espansione di memoria da 16K, registratore a cassette, manuali inglese ed italiano, software, il tutto **vendo** a L. 180.000.
Alberico Di Profio - Via Sirtori 16 - 10148 Torino - Tel. 298458.

Scambio **programmi** per **ZX Spectrum**. Inviatemi la vostra lista ed vi invierò la mia.
Marcello Pellacani - Via G. Guarini 41 - 41100 Modena - Tel. 243649.

Vendo ZX Spectrum 48K + alimentatore rebit + cavetti + manuale italiano e inglese + libro 77 programmi + cassetta dimostrativa e 5 giochi a L. 450.000.
Luca Paladini - Via G. Prina 24 - 00139 Roma - Tel 06/8188579.

Disponendo di una vasta gamma di **software** per **ZX Spectrum** vendo cassette 10 programmi a L. 15.000. Specificare i giochi e utilità desiderate. Spedizione entro 24 ore. Telefonare ore pasti.
Fabio D'Amico - Via 5 giornate 14 - 20025 Legnano (MI) - Tel. 0331/595581.

Scambio **software** per **ZX Spectrum**. Inviare elenco, rapida risposta.
Stefano Orlando - Via Petrarca 153 - 20099 Sesto S. Giovanni (MI) - Tel. 02/2405323.

Programmi per **Spectrum** circa 500 vendo a bassissimo prezzo. Vi prego inviare L. 500 in francobolli per ricevere le liste.
Loredani Raponi - Via Alberto da Giussano 5 - 00176 Roma.

Cambio programmi per **ZX Spectrum.** Oltre 350 titoli a disposizione. Inviare lista e L. 500 per ricevere dettagliate istruzioni. Scrivere solo se avete almeno 80 titoli a disposizione.
Luigi Ballestin - Via M. della Libertà 367/11 - 18038 Sanremo (IM).

Vendo per **ZX Spectrum** 48K 21 favolosi programmi in LM a L. 50.000. A chi li acquista tutti 4 in omaggio. Per la lista scrivere o telefonare (dopo le 20).
Massimo Donati - Via Roma 22 - 56045 Pomarance (PI) - Tel. 0588/ 65322.

Cerco programmi per **ZX81 Sinclair** da 1K. Inviare lista con prezzi ragionevoli.
Guido Mantelli - C.so De Gasperi 43 - 12100 Cuneo.

Vendo ZX81 + esp. 16K + 2 volumi + 3 cassette a L. 180.000. Vendo inoltre programmi per Spectrum.
Flavio Bernadotti - Via Roma 10 - 98050 Malfa (ME).

Varie

Vendo XEROX 820 II, 64K Z80A 4MHz, 2 dischi 8" SS/DD, 960K, con stampante OKI 93 L. 6.500.000 (fatturabili).
Telefonare a Paolo - Tel. 02/2154926.

Wanted Cercasi persone che abbiano sviluppato buoni programmi per **VIC 20 e C 64.** Possibilità di acquisto. Il software deve essere originale.
Renato - Tel. 02/8355488 - ore pasti.

Ingegnere specialista SW per ingegneria civile per **CBM 64, Texas TI 99/4A, Olivetti**, vende programmi calcolo su nastro/disco con informazioni dettagliate e listino completo di SW gestionale.
Mario Siviero - Via Belvedere 101 - 80127 Napoli - Tel. 081/640854.

Vendo cassette vergini C-10 con nastro speciale per computer L. 1.500 + SS contrassegno e diskette 5" 1/4 per CBM-64 L. 5.000 cad. + SS contrassegno.
Lorenzo Fontani - Via Cavour 33 - 53100 Siena - Tel. 0577/284229.

Acquisto le seguenti riviste: **Bit (dal N.1 al N.45), MC (1-26), Personal Software (1-13)**, riviste estere e manuali-libri per l'uso e la programmazione dei personal, in particolare per l'Apple II.
Fonsato - Via Verzuolo 49 - 10139 Torino.

Vendo software di qualsiasi marca.
Roberto Ponzoni - Tel. 02/8266688.

Cambio/vendo software per **Sharp MZ80B** (ing. civile, data base, word processing, utility). Contatto anche possessori di Sharp MZ80K. Converto i programmi per entrambe le macchine con apposita routine.
Stefano Lazzaro - Via Monte Sabotino 2 - 35141 Padova - Tel. 049/22675.

Eseguo perfette **traduzioni** di qualsiasi testo **inglese** (manuali, riviste, articoli, programmi, ecc.). Consegne veloci e prezzi competitivi. Telefonare dopo le ore 17.
Angela Valvo - Via Volontari del Sangue 6 - 20092 Cinisello Balsamo (MI) - Tel. 02/6124138.

Sharp PC1500 + stampante + esp. 5K L. 600.000. Cartucce **VIC 20**: Monitor Programmer's Aid + superesp. grafica L. 30.000 cd. Cabinet di espansione con alimentatore autonomo L. 150.000.
Pocket Group - Club utenti Commodore Puglia - Via Amoruso 34 - 70124 Bari - Tel. 080/510322.

Utenti **Shine computer** cerco per scambio programmi ed esperienze. Cerco programma per **RTTY CW** per lo Shine. Scrivere o telefonare dopo le ore 20.
Emanuele Abbate - Via Garibaldi 129 - 97016 Pozzallo (RG) - Tel. 0932/953739.

Sharp 700 e Sega PC 3000. Cerchiamo programmi di ogni genere, scambiamo, acquistiamo. In preparazione un notiziario "Tutto Sharp" e "Tutto Sega PC 3000". Scriveteci.
Francomputer Club - Franco Gonzato - C.so Fogazzaro 174/139 - 36100 Vicenza - Tel. 0444/42678.

PICCOLI ANNUNCI

Annata 1983 **"Personal Software"** cerco max L. 15.000 o numeri sciolti metà prezzo.
Andrea Pini - Via F. Primaticcio 22 - 40129 Bologna - Tel. C51/362400 solo 8.30-9.00 di mattina, feriali.

Acquisterei programmi interessanti di qualsiasi genere **per IBM sistema 23**. Chi ne fosse in possesso scriva a:
Luigi Sellaroli - Via A. Garibaldi 30 - 04010 Giulianello di Cori (LT).

Sharp 700 MZ. È funzionante il **"Club Sharp"**. Nessuna tassa d'iscrizione. Solo amicizia, onde conoscere, sapere, fare reciprocamente con scambi. Disponibili per tutti già 200 programmi. Mandaci i tuoi.
Fanco Gonzato - C.so Fogazzaro 174 - 36100 Vicenza - Tel. 0444/42678.

Vendo Software per Olivetti **M20:** sistemi esperti (intelligenza artificiale), giochi, gestionale, word processing, software tecnico, grafica interattiva. Richiedete l'elenco gratis. Prezzi bassi.
Elsa Ochetta - Via Beldi 19 - 28068 Romentino (NO).

Vendo Casio FP 1100, 112K 48 ROM, monitor colori e 2 software autodidattici, unità a dischi 320x320 (notevole l'espansione e la grafica) usato pochissimo.
Ruggero Gualtieri - Via Luigi Sacco 3 - 20146 Milano - Tel. 02/437619.

Stampante nuovissima in imballo originale con pacco carta, vendo solo L. 450.000 tratt. Modello Seikosha **GP80N, 80** col. grafica. Ottima per Home computer.
Antonello Stanzione - Via Duca Guglielmo 13 - 84100 Salerno - Tel. 089/392694.

Cerco **possessori Oric 1** per scambio programmi su cassette e listati. Chi ne fosse interessato scriva o telefoni dopo le ore 19.
Giacomo Buratti - Via Metastasio 4 - 20098 San Giuliano Milanese (MI) - Tel. 02/9844350.

Cambio bellissimi **programmi** in **BASIC** e in linguaggio macchina con numeri arretrati di Personal Software. Possiedo programmi di tutti i generi. Scrivere o telefonare solo Lunedì dalle 14.30 alle 15.00.
Alessandro Poletti - P.zza Gino Lucetti 2 - 54031 Avenza (MC) - Tel. 53664.

Vendo programma totocalcio per sviluppare sistemi integrali o ridotti con limitazioni sulle colonne e in più un programma per il controllo automatico dei punti. Tutto a L. 20.000.
Riccardo Campo - Tel. 011/252543 - Torino.

Vendo e/o cambio programmi in **BASIC o CP/M** per personal computer Alphatronic P2. Per accordi telefonare ore serali.
Amedeo Di Salvatore - Tel. 0775/857479 - Frosinone.

Per **compravendita** (su base nazionale) **software**. Scrivere dettagliando offerte e/o richieste. Materiale non richiesto sarà cestinato.
Studio Veltri - Via D' Antona 20 - 80131 Napoli.

Cambio/vendo monitor a colori Hantarex mai usato : **CT9001SR14**" a L. 850.000 e scambio programmi per Apple II.
Tiziano Maffei - Il Podere Loc. S. Giovanni - 57037 Portoferraio (LI) - Tel. 0565/916566.

Vendo due bellissimi giochi per **Atari 400-800** su cassetta: "Black Hole" e "Battaglia aereonavale", 13K di memoria. Una cassetta con due giochi L. 60.000. Cerco Ataristi per scambi programmi ed idee.
Cristiano Fanucci - Via Spartaco 29 - 40138 Bologna - Tel. 051/533343.

Vendo videogame **Atari VCS 2600** quasi nuovo, in ottimo stato, più cassette STARMASTER (Activision), Breakout e combat a L. 200.000.
Cristoforo Grillo - Via G. Zanella 19 - 35020 Albignasego (PD) - Tel. 049/711351.

A Torino è nata **l'associazione CEDAT**. Amici del computer unitevi per scambio idee e programmi di tutte le marche. Per informazioni rivolgersi in Via Susa 14 - Torino - Tel. 011/7492827.

Cerco disperatamente **modulo SSS TI** Extended BASIC. Telefonare dalle ore 12 alle ore 16.
Sergio Viganò - Via C. Gabriele 3 - 20050 Camparada (MI) - Tel. 039/6980357.

Vendo Atari computer sistem + seguenti cassette: Pole Position, Tennis, Basketball, Bowling, 3D Tic Tac Toe, Soccer's Apprentice. Comprato tutto Natale 83. Prezzo L. 150.000. Telefonare ore pasti.
Nicola Neri - Tel. 0744/418114 - Terni.

Cerco listati programmi per **Olivetti M20** di giochi o grafica. Telefonare ore pomeridiane o inviare per posta. Disposto anche a comperare i listati.
Antonio Moro - Via A. De Gasperi 56 - 36022 Cassola (VI) - Tel. 0424/83027.

Compro solo se vera occasione **programmi in blocco** di ogni tipo. Inviatemi subito la vostra lista. Non perdete questa occasione.
Piero Zambrini - Via Ridolfino Venuti 14 - 00162 Roma - Tel. 06/8320808.

Offro anche separatamente videogioco **Atari CX 2600** tripli comandi 5 cassette a L.320.000. Telefonare dopo le 20.00.
Davide Carancini - C.so Laghi 81/77 - 10090 Buttigliera A. (TO) - Tel. 011/9322242.

Vendo progr. per **PC IBM** "gestione c/c" inedito, elaborato per una media azienda, fornisce estr. c., s. contabile, valuta, S.F.B.,interessi, ecc. L. 100.000 su dischetto con aiuto personalizzaz.
Dr. Alessandro Faenzi - Via Kennedy 5 - 06083 Bastia Umbra (PG) - Tel. 075/8002227.

Vendo oscilloscopio "doppia traccia" marca **Triomod**. CS/1554 con accessori e manuale L. 600.000. **Vendo corso TV scuola "Radio Elettra"** solo parte pratica per costruzione TV b/n 12" ancora imballato L. 100.000. Dispongo altri strumenti per bassa frequenza.
Giorgio Invernizzi - Via Pastonchi 5 - 22063 Cantù (CO) - Tel.707162.

Vendo Atari 400 + registratore inoltre vendo, anche separatamente, molti programmi provenienti dall'America.
Luigi Stadio - Via M. Schipa 61 - 80122 Napoli - Tel. 081/664626.

Cerco, cambio vendo software vario per Sharp NZ80A. Inviare lista o scrivere a:
Davide Ardizzola c/o "La Base" - Via Roma - 18016 S. Bartolomeo (IM) - Tel. 0183/401830.

Vendo Atari VCS L. 170.000 intrattabili. 13 cassette in blocco a L. 450.000 oppure 3/4 a prezzo originale.
Francesco Di Pede - Titignano 9 - 56023 Titignano (PI) - Tel. 050/7704460.

Compro espansione 16K RAM e cartuccia gioco scacchi. Possibilmente zona nord-Sardegna.
Maurizio Iannucci - Via E. Vanoni 5 - 07041 Alghero (SS).

Cerco TRS-80ntisti preferibilmente nella zona di Catania per scambio informazioni e idee. Scrivete risponderò a tutti.
Angelo Castorina - P.zza Trento 2 - 95128 Catania - Tel. 095/446506.

Possedete la **scheda grafica** di **"Nuova elettronica"**, volete entrare nel mondo del CP/M ? Vendo Epron e relative modifiche con dischetto applicativo.
Antonio Romano - Via delle Conce 32r - 50122 Firenze - Tel. 055/663859.

Vendo microcomputer Z-80 di "Nuova elettronica" con floppy disk drive e monitor nec a fosfori verdi a 48K di RAM, BASIC/assembler-disassembler con oltre 800K di programmi applicativi vari a L. 1.450.000 tratt.
Dario Melpignano - V.le Corsica 3 - 20133 Milano - Tel. 02/720553.

Vendo Acorn Atom perfetto poco usato espanso 12 + 16K L. 450.000. Regalo manuale italiano (ed. muzzi) e programmi originali giochi/utilità oppure cambio con unità dischi 1541XCBM64.
Roberto Fermo - Via F. Crispi 26 - 80121 Napoli - Tel. 081/667177.

Vendo p. computer FP II completo di: tastiera esterna, interfaccia per 2 drive, drive slim line, registratore, molte cassette gioco e utility su disco, joystick e tutti i manuali, garanzia intatta prezzo L. 1.500.000.
Giuseppe Culmone - Via 2/F 11 - 90011 Alcamo (TP) - Tel. 0924/29044 - 29158.

Vendo computer NE schede LX 392 L. 80.000, LX386 L. 150.000, LX 382 L. 80.000, monitor fosfori verdi 12" L. 20.000.
Renato Severin - Via Casaria 23 - 31030 Biancade (TV) - Tel. 0422/790136.

Vendo interfaccia joystick **"Tecnolek"** programmabile, joystick anatomico con 2 pulsanti sparo e cassetta dimostrativa originale nuovo di pacca a L. 105.000.
Alan Sartirani - P.zza Libertà 3 - 24044 Dalmine (BG) - Tel. 562380.

Vendo corso completo **di informatica** dell'Arexa comprendente anche corso ad alto livello di Cobol e BASIC. Vendo anche PB 100 + espan. 1Kb + interfaccia registratore; tutto con 3 mesi di vita.
Luca Beccati - Via Bandiera 14 - 44034 Copparo (FE) - Tel. 0532/862022.

Desideriamo formare un **computer club** (commodore - sinclair) per scambio software chi volesse entrarne a far parte scriva a questo indirizzo e riceverà chiarimenti via posta.
Matteo Linares - Via R. Pilo 4 - 91025 Marsala - Tel. 0923/959922.

PICCOLI ANNUNCI

PERSONAL SOFTWARE

Sei un lettore di PERSONAL-SOFTWARE e vuoi entrare in contatto con tutti gli altri lettori per comperare, cambiare o vendere software? Spedisci questo tagliando a Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 Milano

Nome .... Cognome ....

Via .... C.A.P. ....

Città .... Tel. ....

CEDOLA DI COMMISSIONE LIBRARIA

PERSONAL SOFTWARE

Da inviare a Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 Milano

Nome Cognome

Indirizzo

Cap. Città Provincia

Partita I.V.A. (indispensabile per le aziende)

Si richiede l'emissione della fattura

Inviatemi i seguenti libri:

Codice Libro	Quantità	Codice Libro	Quantità	Codice Libro	Quantità	Codice Libro	Quantità

□ Pagherò al postino il prezzo indicato + L. 2.000 per contributo fisso spese di spedizione.

□ Allego assegno n° .... di L. ....

□ Non abbonato □ Abbonato sconto 20% □ l'Elettronica □ Elettronica Oggi □ Automazione Oggi □ Elektor
□ Informatica Oggi □ Computerworld □ Bit □ Personal Software □ Strumenti Musicali □ Videogiochi

Data .... Firma ....

UNA PUBBLICAZIONE
DEL GRUPPO EDITORIALE JACKSON

PERSONAL SOFTWARE

ANNO 3 N. 19 GIUGNO-LUGLIO 1984

DIRETTORE RESPONSABILE: Giampietro Zanga
CAPO REDATTORE: Lorenzo Barrile
DIRETTORE TECNICO: Riccardo Paolillo
GRAFICA E IMPAGINAZIONE: Renata Lavizzari
HANNO COLLABORATO A QUESTO NUMERO:
W. Sini, R. Gallieni, C. Giavani,
F. Ghedini, A. Guida, L. Marras,
G. Morpurgo, C. Cappelli, F. Cerulo,
S. Borsani, M. Giacobazzi, M. Lenzi,
M. Spero.
UFFICIO ABBONAMENTI: Tel. 6880951/2/3/4/5
USA: G.E.J. Publishing Group, Inc. a Subsidiary of - Gruppo Editoriale Jackson 1143 Quince Avenue - 94087 Sunnyvale (CA) Tel. (408) 7730103 - Telex: 4995972
GRAN BRETAGNA: G.E.J. Publishing L.t.d. 187 Oxford Street London WIR1AJ Tel. (01) 4392931 - Telex: (051) 21248
PUBBLICITA': Concessionario per l'Italia e l'Estero Reina S.r.l. Via Washington, 50 - 20146 Milano Tel. (02) 4988066/7/8/9/060 (5 linee r.a.) Telex: 316213 REINA I
PUBBLICITA' PER USA E CANADA: International Media Marketing 16704 Marquardt Avenue, P.O. Box 1217 Cerritos (CA) 90701 - (213) 926-9552
FOTOCOMPOSIZIONE: LINEACOMP S.r.l. Via Rosellini, 12 - 20124 Milano
STAMPA: Reweba - Brescia
Concessionario esclusivo per la diffusione in Italia e all'Estero: SODIP - Via Zuretti, 25 - 20125 Milano
AUTORIZZAZIONE ALLA PUBBLICAZIONE: Tribunale di Milano n. 69 del 20/2/1982
Spedizione in abbonamento postale Gruppo III/70
Prezzo della rivista L. 4.000. Numero arretrato L. 8.000.
Abbonamento annuo (10 numeri) L. 34.000; per l'Estero L. 51.000
I versamenti vanno indirizzati a: Gruppo Editoriale Jackson Via Rosellini, 12 - 20124 Milano - mediante emissione di assegno bancario, cartolina vaglia o utilizzando il C/C postale numero 11666203.
Per i cambi di indirizzo, indicare, oltre naturalmente al nuovo, anche l'indirizzo precedente, ed allegare alla comunicazione l'importo di L. 500, anche in francobolli.

Il Gruppo Editoriale Jackson è iscritto nel registro Nazionale della stampa al n. 117 - Vol. 2 - Foglio 129 in data 17-8-1982

GRUPPO EDITORIALE JACKSON S.r.l.

DIREZIONE, REDAZIONE, AMMINISTRAZIONE: Via Rosellini, 12 - 20124 Milano - Telefoni: 68.80.951/2/3/4/5
SEDE LEGALE: Via G. Pozzone, 5 - 20121 Milano
DIREZIONE EDITORIALE: Giampietro Zanga e Paolo Reina
COORDINAMENTO EDITORIALE: Daniele Comboni
DIREZIONE AMMINISTRATIVA: Giuliano Di Chiano

OFFERTISSIMA – EXELCO –

La EXELCO vi propone due vantaggiosissime combinazioni COMMODORE
Affrettatevi è un'OFFERTA irrepetibile!!!





Combinazione n° 1

n° 1 COMMODORE C64	L. 699.500
n° 1 Registratore «MAXTRON»	L. 82.000
n° 1 Libro «C64 EXPOSED»	L. 24.000
n° 1 Libro «Impariamo ad usare il personal computer	L. 25.000
n° 1 Cassetta Software «Garden Wars» della JCE	L. 20.000
Totale	L. ~~850.000~~

A SOLE L. 750.000
IVA INCLUSA

Combinazione n° 2

n° 1 COMMODORE VIC 20	L. 149.000
n° 1 JOYSTIC «TRIGA»	L. 31.000
n° 1 Libro «Impariamo ad usare il personal computer	L. 25.000
n° 1 Cassetta Software «ADVENTURELAND»	L. 49.000
n° 1 Cassetta Software «RIDE ON FORT KNOX»	L. 49.000
Totale	L. ~~303.000~~

A SOLE L. 258.000
IVA INCLUSA

Descrizione	Q.tà	Prezzo unitario	Prezzo Totale
COMBINAZIONE 1: n° 1 COMMODORE C64, n° 1 Registratore «MAXTRON», n° 1 Libro «C64 EXPOSED», n° 1 Libro «Impariamo ad usare il personal computer», n° 1 Cassetta Software «Garden Wars» JCE		L. 750.000	
COMBINAZIONE 2: n° 1 COMMODRE VIC 20, n° 1 JOYSTIC «TRIGA", n° 1 Libro «Impariamo ad usare il personal computer», n° 1 Cassetta Software «ADVENTURELAND», n° 1 Cassetta Software «RIDE ON FORT KNOX»		L. 258.000	

Desidero ricevere il materiale indicato nella tabella, a mezzo pacco postale contro assegno, al seguente indirizzo:

Nome

Cognome

Via

Città

Data C.A.P.

Desidero ricevere la fattura SI NO

Partita I.V.A. o, per i privati Codice Fiscale

PAGAMENTO:
A) Anticipato, mediante assegno circolare o vaglia postale per l'importo totale dell'ordinazione.
B) Contro assegno, in questo caso, è indispensabile versare l'acconto di Lire 50.000 mediante assegno circolare o vaglia postale. Il saldo sarà regolato contro assegno.
AGGIUNGERE: L. 5.000 per contributo fisso. **I prezzi sono comprensivi di I.V.A.**

EXELCO Via G. Verdi, 23/25
20095 - CUSANO MILANINO - Milano

SI ACCETTANO FOTOCOPIE DI QUESTO MODULO D'ORDINE

NON PERDETE QUESTE OCCASIONI!!!!

ONE HUNDRED STEPS TO LONDON

CONCORSO per tutti i SINCLAIRisti



COME FUNZIONA IL CONCORSO?

Prima di entrare nel dettaglio del Regolamento che per altro consigliamo di leggere attentamente, descriviamo il meccanismo di questo strabiliante concorso, organizzato dalla J.C.E.
Ci preme innanzitutto chiarire che la Giuria è il pubblico, ossia i lettori di Sperimentare con l'Elettronica e il Computer, e ciò sottolinea la filosofia di dialogo e di fiducia, in cui la redazione non si pone, come in altri casi spesso avviene, nella presuntuosa posizione di infallibilità!
Ogni mese a partire dal numero di Giugno 1984, la rivista pubblicherà i quattro migliori programmi pervenuti in redazione, e giudicati dalla nostra commissione tecnica. I lettori troveranno quindi i listati di questi quattro programmi ed un tagliando sul quale scriveranno il titolo del programma che sarà parso più meritevole, per qualsiasi motivo.
Fra tutti i lettori che avranno inviato il tagliando sarà estratto, ogni mese, un computer Sinclair Spectrum 16 K!!!
Questa procedura si ripeterà per sei mesi, quindi sui numeri di Giugno, Luglio/Agosto, Settembre, Ottobre, Novembre e Dicembre, quindi ogni mese saranno pubblicati quattro programmi, il tagliando da spedire e fra i tagliandi sarà estratto uno Spectrum 16 K.
Alla fine di questa prima fase usciranno, dunque sei programmi classificati al primo posto in ciascuna delle sei "batterie".
Noi li sottoporremo al giudizio dei lettori, pubblicando sul numero di Marzo 1985 un nuovo tagliando sul quale andranno indicati, in ordine di preferenza, tutti i sei programmi. I primi tre saranno premiati, e fra i lettori sarà estratto di nuovo uno Spectrum 16 K.
Mica male, eh, che ne dite?
Il regolamento è riportato sulla rivista "Sperimentare con l'Elettronica e il Computer".

1° CONCORSO SINCLUB 1984-85

I Sinclair Club e i Sinclairisti sono spesso vere e proprie fonti di idee per il miglior impiego del nostro beneamato spectrum; è un peccato che la maggior parte di queste idee restino nel cassetto, o nel migliore dei casi vengano resi noti alla ristretta cerchia degli amici.
Uno degli scopi del Sinclub è proprio quello di dare le ali a chi le merita. Da qui l'idea di un concorso a premi, organizzato dalla **J.C.E.**, aperto a tutti. Il concorso inizierà il **1° Aprile** e terminerà il **31 Dicembre 1984**; tutti potranno inviare alla redazione di **Sperimentare con l'Elettronica e il Computer** entro il suddetto periodo loro programmi più interessanti ed originali.

I programmi devono essere memorizzati su cassetta e devono essere spediti alla redazione di Sperimentare con l'Elettronica e il Computer al seguente indirizzo:

J.C.E. - CONCORSO SINCLUB
VIA DEI LAVORATORI, 124
20092 CINISELLO BALSAMO (MI)